



GMD Research Series

GMD –
Forschungszentrum
Informationstechnik
GmbH

Andreas Becks

Visual Knowledge Management with Adaptable Document Maps

© GMD 2001

GMD – Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
<http://www.gmd.de>

In der Reihe GMD Research Series werden Forschungs- und Entwicklungsergebnisse aus der GMD zum wissenschaftlichen, nichtkommerziellen Gebrauch veröffentlicht. Jegliche Inhaltsänderung des Dokuments sowie die entgeltliche Weitergabe sind verboten.

The purpose of the GMD Research Series is the dissemination of research work for scientific non-commercial use. The commercial distribution of this document is prohibited, as is any modification of its content.

Anschrift des Verfassers/Address of the author:

Andreas Becks
Institut für Angewandte Informationstechnik (FIT)
GMD – Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
E-mail: andreas.becks@gmd.de

Die Deutsche Bibliothek - CIP-Einheitsaufnahme:

Becks, Andreas:

Visual Knowledge Management with Adaptable Document Maps /
Andreas Becks. GMD – Forschungszentrum Informationstechnik GmbH. -
Sankt Augustin : GMD – Forschungszentrum Informationstechnik, 2001
(GMD Research Series ; 2001, No. 15)
Zugl.: Aachen, Techn. Hochsch., Diss., 2001
ISBN 3-88457-398-5

ISSN 1435-2699

ISBN 3-88457-398-5

Abstract

Analyzing, structuring and organizing documented knowledge is an important aspect of knowledge management. In literature so-called document maps have been proposed for visualizing the semantic similarity structure of a corpus of documents. So far, however, a method which is specifically designed for typical document analysis tasks in knowledge management – along with an application-oriented evaluation – was missing. Based on an empirical task-model this work presents an adaptable framework for generating document maps. The benefits of this framework are a flexible combination of similarity- and topology-preserving visualization methods and the exchangeability of the component for assessing the similarity of documents. An extension of the basic method allows the analyst to adapt the map's generation by incorporating a personal analysis interest. The interactive document map system DocMINER implements the developed method and comprises additional map-oriented tools for analyzing text collections. Industrial and scientific case studies help to better understand whether the proposed graphical overview can be effectively applied to real-world problems. Moreover, a comparative empirical study in a laboratory setting evaluates the document map concept against an alternative text-access interface.

ACM CLASSIFICATION

Subject Descriptors: H.3.3 Information Search and Retrieval (Clustering); H.5.2 User Interfaces (Evaluation, Methodology, Graphical User Interfaces, User-centered Design); H.4.m Information Systems Applications (Miscellaneous)

General Terms: Algorithms, Design, Experimentation, Human Factors, Measurement

Additional Key Words and Phrases: Knowledge Management, Text Corpus Analysis

Kurzfassung

Dokumentiertes Wissen zu analysieren, zu strukturieren und zu organisieren ist ein wichtiger Bestandteil des Wissensmanagements. In der Literatur sind sogenannte Dokumentenlandkarten zur Visualisierung der inhaltlichen Ähnlichkeitsstruktur eines Textkorpus vorgeschlagen worden. Allerdings stand bislang eine an den typischen Aufgaben der Dokumentenanalyse im Wissensmanagement orientierte Methodik und eine entsprechende applikationsspezifische Evaluierung aus. Ausgehend von einem empirischen Aufgabenmodell wird daher in dieser Arbeit ein adaptierbares Rahmenmodell zur Generierung von Dokumentenlandkarten vorgestellt. Dieses bietet u.a. eine flexible Verknüpfung ähnlichkeits- und topologieerhaltender Visualisierungsmethoden und sichert die Austauschbarkeit der Komponente für die Bewertung der Textähnlichkeit. Eine Erweiterung dieser Basismethode erlaubt es dem Analysten, die Berechnung der Karte durch die Formulierung eines Analyseinteresses zu adaptieren. Das interaktive Dokumentenlandkarten-System DocMINER realisiert die vorgeschlagene Methodik und bietet zusätzliche, auf die Kartendarstellung abgestimmte Funktionen für die Analyse von Textsammlungen an. Fallstudien in industriellen und wissenschaftlichen Kontexten helfen zu verstehen, ob die vorgeschlagene grafische Übersicht für praktisch relevante Problemstellungen effektiv anwendbar ist. Eine vergleichende empirische Untersuchung in einer kontrollierten Laborumgebung bewertet zudem das Konzept der Karten gegenüber einer alternativen Zugriffsmethode für Textsammlungen.

ACM KLASSEIFIKATION

Deskriptoren: H.3.3 Informationssuche und -retrieval (Clustering); H.5.2 Benutzerschnittstellen (Evaluation, Methodologie, Grafische Benutzerschnittstellen, Benutzerzentriertes Design); H.4.m Informationssysteme – Anwendungen (Verschiedenes)

Allgemeine Begriffe: Algorithmen, Design, Experimente, Menschliche Faktoren, Messung

Zusätzliche Schlagwörter: Wissensmanagement, Textkorporusanalyse

Acknowledgements

This thesis was created during my time as a PhD-student at the Graduate School on 'Informatics and Engineering' (funded by the Deutsche Forschungsgemeinschaft) at the Department of Information Systems (Informatik V) and during my employment at the Institute for Applied Information Technology of GMD German National Research Center for Information Technology. Parts of the research reported in this thesis were conducted in the collaborative research center "Medien und kulturelle Kommunikation" (Media and Cultural Communication, FK/SFB 427). I thank Matthias Jarke, my thesis advisor, for making possible this work and for his trust, support and feedback. Thanks also to Ludwig Jäger who provided valuable feedback on the concept of document maps from a cultural science point of view.

The case studies performed in this work would not have been possible without the support of many people. Particularly I would like to thank Jörg Köller, Michael Host, and Ralf Klamma. The members of the special interest group 'Technical Documentation' of Aachen's regional industry club REGINA, and participants of the 'tekom Frühjahrstagung 99', especially Friedrich W. Fabri, contributed to the work by many fruitful discussions on the issue of managing product documentation. Kai-Hendrik von der Aa, Erwin Flender, and the members of the work group 'Terminology' of the collaborative research center 'Media and Cultural Communication' provided rich material and helpful feedback.

I also received great support for the empirical investigation: I thank Eva-Maria Jakobs, Jörg Jost, Ralf-Dieter Hilgers and his staff, especially Ralf Minkenberg, for many discussions regarding the design and statistical evaluation of the study, Mareike Schoop and Christoph Quix for their organizational support, the staff of the system administration group of the department of computer science for the technical assistance, and of course the students who volunteered as test persons.

My master students Carsten Tusk and Christian Seeling provided many ideas from which this work could benefit. Christian Seeling has also implemented major parts of the developed system during his time as collegiate assistant. Thanks go also to my colleagues at RWTH Aachen and GMD-FIT, especially to Jürgen Rack and Stefan Sklorz, for countless helpful discussions on the subject of my work. I thank Uwe Tüben, Carsten Tusk, and Christian Seeling for grammatical and stylistic language corrections (of course I am to blame for remaining errors).

Last but not least, I thank my parents Ursula and Christian Becks for their support and for making possible my education, and my wife Kathrin for her love, her endless patience, and her encouragement.

Aachen, May 2001

Table of Contents

1	INTRODUCTION	1
1.1	Problem Description: Analyzing Text Corpora in Knowledge Management	1
1.2	Solution Idea: Providing Visual Access to a Text Corpus	3
1.3	Research Methods and Contributions	4
1.4	Thesis Outline	5
2	A TASK MODEL FOR TEXT CORPUS ANALYSIS IN KNOWLEDGE MANAGEMENT	9
2.1	General Definitions and Requirements	10
2.2	Goal-Directed versus Explorative Document Access	11
2.3	A Brief Review of Task Models	12
2.3.1	Overview and Classification of Prominent Task Models	12
2.3.2	Detailed Presentation of Selected Task Models	14
2.4	A New Basic Task Model Derived from Literature	15
2.4.1	Comparison of Selected Task Models	16
2.4.2	A New Basic Task Model	18
2.5	Enriching the Basic Model: Analysis of Real-World Tasks	19
2.5.1	Extracting Corpus Analysis Tasks from Practical Applications	19
2.5.2	Optimizing the Basic Model	21
2.6	The Resulting Task Model	22
2.6.1	Definition of Criteria and Values	22
2.6.2	Application of the Model	25
2.6.3	Describing and Comparing Tasks in the Model: An Example	25
2.6.4	Reflecting the Task Model	27
3	METHODS FOR TEXT REPRESENTATION AND COMPUTING DOCUMENT SIMILARITY	29
3.1	Vector-Based Methods from Information Retrieval	29
3.1.1	Indexing Terms and the Vector Space Model	29
3.1.2	Latent Semantic Indexing and Concept Vectors	30
3.2	Text Analysis Based on Explicitly Defined Relationships	30
3.2.1	Citation Processing	30
3.2.2	Hypertext Linkage	31
3.2.3	Information Retrieval and XML	31
3.3	Text Understanding and Domain Knowledge	32

3.3.1	Structured Text Representations	32
3.3.2	Information Extraction	32
3.3.3	Textual Case-Based Reasoning	32
3.3.4	A Knowledge-Based Approach for Comparing Medical Abstracts	33
4	METHODS FOR VISUALIZING SIMILARITY DATA	35
4.1	Multidimensional Scaling and Related Techniques	35
4.1.1	Metric and Non-Metric Multidimensional Scaling Methods	36
4.1.2	Multidimensional Scaling versus Factor Analysis	37
4.1.3	Force-Directed Placement	37
4.1.4	Geometric Scaling	38
4.1.5	Scatter Plots for Visualizing Scaled Data Sets	38
4.2	Self-Organizing Feature Maps and their Visualization	39
4.2.1	The Model of Self-Organizing Feature Maps in a Nutshell	39
4.2.2	Properties of the Model and Comparison with Scaling Methods	40
4.2.3	Visualization of Self-Organizing Maps	40
5	STRUCTURING AND VISUALIZATION TECHNIQUES FOR DOCUMENT COLLECTIONS	43
5.1	Semantic Structure for Guiding Information Access	43
5.1.1	Focus of Interest	44
5.1.2	Background: Clustering in Information Retrieval	45
5.2	Structuring Document Collections for Exploration	46
5.2.1	Categorization and Dynamic Clustering	46
5.2.2	Hypertext Structures	47
5.3	Visualizing the Structure of a Text Corpus: An Overview	49
5.3.1	Visualizing Hypertext Structures	49
5.3.2	Visual Information Retrieval Interfaces	50
5.3.3	Displaying the Structure of Query Result Sets	51
5.4	Document Maps and Landscapes	51
5.4.1	BEAD	52
5.4.2	SPIRE	52
5.4.3	VXINSIGHT	53
5.4.4	STARLIGHT / TRUST	54
5.4.5	KNOWLEDGE GARDEN	55
5.5	Self-Organizing Document Maps	55
5.5.1	Category Maps	55
5.5.2	Classification Aided by Self-Organizing Document Maps	58
5.5.3	WEBSOM	60
5.6	Evaluations of Graphical Overviews for Searching Texts	62
5.7	Discussion in the Context of Text Corpus Analysis	63
5.8	Motivation and Goals of the Own Approach	67
5.8.1	Visualization Principle and its Degree of Granularity	67
5.8.2	A Modular Approach for Structuring Specialized Text Collections	68

5.8.3	Application-Oriented Evaluation	68
5.8.4	Extension: An Adaptable Document Map Approach	69
5.8.5	Summary of the Discussion and Overview of Contributions	69
6	A BASIC FRAMEWORK FOR DOCUMENT MAPS	71
6.1	Design of the Basic Framework: Overall Architecture	71
6.1.1	Variable Document Analysis Module	71
6.1.2	Variable Spatial Scaling Module	72
6.1.3	Topology Preserving Mapping and Visualization Module	73
6.2	Realization of the Document Map Approach	74
6.2.1	Document Analysis Module	74
6.2.2	The Spatial Scaling Module	80
6.2.3	Topology Preserving Mapping and Visualization Module	85
6.2.4	Summary of the Basic Map Generation Process	89
6.3	Interpreting a Document Map	90
6.4	Setting Parameters for Generating Document Maps	92
6.4.1	Performance of the Spatial Scaling Module	93
6.4.2	Setting Parameters for the Topology Preserving Mapping	95
6.4.3	Influencing Structures by Weighting Document (Dis-)Similarities	96
6.4.4	Excursus: Measuring the Semantic Document Space's Structure	97
6.4.5	Parameters of the Document Analysis Module	100
6.4.6	Complexity of the Document Map Approach	102
6.5	Experiment: Structuring Newsgroup Articles	103
7	DOCMINER – AN INTERACTIVE MAP-CENTERED CORPUS ANALYSIS TOOL	107
7.1	Objectives of the System Development	107
7.2	Overall Architecture	108
7.3	Generating Document Maps: Project Workbench	109
7.4	Working with a Map: Interactive Features of the System	110
7.4.1	Basic Document Map Interaction	111
7.4.2	Characterizing Document Groups: Term Profiles and Summaries	112
7.4.3	Coupling a Query Interface with the Map Display	114
7.5	Usability Criteria Met by the Prototype	115
7.6	A Sample Session	117
8	CASE STUDIES	121
8.1	Case Study I: Structuring Use Cases in Software Engineering	121
8.1.1	Managing Collections of Requirement Scenarios	122
8.1.2	Use Cases for Standardizing Open Simulation Environments	125
8.1.3	A Semantic Map of Use Cases in CAPE OPEN	128
8.1.4	Summary and Conclusion	135

8.2 Case Study II: Managing Product Documentation of Software Vendors	136
8.2.1 Managing Product Documentation – Some Scenarios	137
8.2.2 Checking the Quality of User Manuals	139
8.2.3 Inspecting a Complete Product Documentation Collection	143
8.2.4 Conclusion	144
8.3 Case Study III: Supporting Terminology Work in a Cultural Science Project	147
8.3.1 Knowledge Organization in a Cultural Science Project	147
8.3.2 Cooperative Document Analysis with Document Maps	149
8.3.3 Some Results of the Cooperative Document Analysis	154
8.3.4 Evaluating the Support Offered by Document Maps	155
8.3.5 Conclusion and Outlook	156
8.4 Summary of Results	157
9 TASK-ADEQUACY OF DOCUMENT MAPS: A COMPARATIVE LABORATORY STUDY	159
9.1 Research Question and Objectives	159
9.2 Experimental Design	160
9.2.1 System Support: Document List and Document Map	160
9.2.2 Type of Measurement	163
9.2.3 Cornerstones of a Test Design Inspired by Practical Scenarios	163
9.2.4 Defining Measures for Assessing a System's Task-Adequacy	164
9.2.5 Variables and Influence Factors	167
9.2.6 Designing Practically Relevant Tasks	169
9.2.7 Formal Hypotheses	177
9.3 Preparing and Conducting the Experiment	180
9.3.1 Pilot Study	180
9.3.2 Generating the Document Map for the Experiment	180
9.3.3 Introductory Lecture and Practical Training	181
9.3.4 Course of the Experiment	181
9.4 Results of the Experiment	182
9.4.1 Statistical Test Procedures Used	183
9.4.2 Statistical Characterization of Samples	183
9.4.3 Results of the Tasks	185
9.4.4 Results of the Qualitative Questionnaires	187
9.4.5 Observed Correlations Between Variables	191
9.4.6 Comments on Systems by the Test Subjects	192
9.4.7 Results of the Interaction Protocols	192
9.5 Interpretation and Conclusion	194
9.5.1 Interpreting the Results for Each Task	194
9.5.2 Overall Conclusion	196
10 EXTENSION OF THE BASIC FRAMEWORK: INCORPORATING ADAPTABILITY	199
10.1 A Rule-Based Approach for Incorporating Adaptability	199
10.2 General Design Decisions	200
10.2.1 Integration into the Basic Framework	201

10.2.2	Design of the Rule Approach	202
10.3	Moving Document Representatives by Attractive Forces	203
10.3.1	The Attractor Concept	203
10.3.2	Realization of the Proposed Attractor Approach	204
10.3.3	Illustration of Object Movement Defined by Attractors	206
10.4	Overview of the Rule Design	208
10.4.1	Basic Modeling Elements	208
10.4.2	Rule Types	210
10.5	Excursus: Techniques for Implementing Rules	212
10.5.1	Fuzzy Set Theory	213
10.5.2	Part of Speech Tagging	214
10.6	Realization of the Proposed Rule Base Approach	215
10.6.1	Concepts and Attribute-Value Pairs	215
10.6.2	Matching Modeling Elements Against Documents	217
10.6.3	Matching Rules Against Documents	223
10.6.4	Defining the Attractor Strength Matrix on Basis of the Rules	228
10.7	Complexity of the Semantic Refinement Module	230
10.8	Reflection of the General Design	231
10.9	Integration into the DocMINER System	233
10.9.1	Extended Architecture	233
10.9.2	Extended Workspace and Additional Tools	233
10.10	Experiments	235
10.10.1	Experiment I: Aeronautics Patent Abstracts	236
10.10.2	Experiment II: Airplane Descriptions	241
10.10.3	Conclusion	245
11	CONCLUSION	247
11.1	Summary of Results and Contributions	247
11.2	A Media-Theoretic Perspective on Document Maps	248
11.3	Outlook	248
11.3.1	Extending the Evaluation	249
11.3.2	Possible Improvements of Map Technology and System	249
11.3.3	Further Research Directions	250
REFERENCES		252
A	ADDITIONAL EXPERIMENTAL NEWSGROUP DOCUMENT MAPS	267
B	ORIGINAL TASK SHEETS AND QUESTIONNAIRES FOR COMPARATIVE STUDY	271
C	GUI QUICK REFERENCE SHEETS USED IN COMPARATIVE STUDY	281
D	DOCUMENT MAP USED IN COMPARATIVE LABORATORY STUDY	283

E	PART OF SPEECH TAGS USED BY THE SEMANTIC REFINEMENT MODULE	286
F	RULE BASE SYNTAX OF THE SEMANTIC REFINEMENT MODULE	287
G	SEMANTIC REFINEMENT MODULE: RULE BASES AND ADDITIONAL EXPERIMENTS	291
H	DOCUMENT MAP OF THIS THESIS	296

1 Introduction

Information access is a central research topic with still increasing relevance in the context of a global knowledge and information society. In times where specific information is of highest value for enterprises or organizations, and knowledge becomes the most important production factor, techniques enabling an adequate access to information play the role of a key technology for an appropriate management of knowledge. Often it is not the lack of knowledge sources in a company that is a problem, but the flood of unstructured information. Thus, one important technological aspect of knowledge management is to support companies in their efforts to structure, to condensate and to learn from their documented knowledge assets. This thesis studies the concept of graphically presenting the inherent semantic structure of a text corpus. The aim is to support an effective and productive analysis of corporate document collections. Before the goals of this thesis are described, the following section provides some background for the problem of handling documented knowledge.

1.1 Problem Description: Analyzing Text Corpora in Knowledge Management

Information and knowledge are commonly seen as the most important production factors in modern industry. For many companies their proprietary knowledge is their only strategic advantage, and it is necessary to leverage this valuable resource. Following Aamodt and Nygard [AaNy95], knowledge – as the output of a learning process – is created from information (i.e. interpreted data) by incorporating it into an agent’s reasoning resources. This, however, requires the possibility to adequately access and combine the right information at the right time. Once knowledge is created, it is ready for active use in decision making. But in which forms does this valuable asset exist? Nonaka [NoTa97] distinguishes two epistemological dimensions of knowledge, namely *implicit and explicit knowledge*. Implicit (or tacit) knowledge is personal knowledge that depends on its context and is hard to communicate. As such it is subjective “knowledge in action”, bound to human resources. In contrast, explicit knowledge can be described formally and systematically. It is objective, documented and thus accessible and stored for the company. Implicit knowledge can be passed on by a non-verbal exchange of experiences (a process called *socialization*). Explicit knowledge can be *combined*, thus resulting in new explicit knowledge. By articulating implicit knowledge it becomes explicit and can be encoded, for instance, in documents – a process called *externalization*. Such a documented knowledge store can then support the process of *internalization*, i.e. the transformation of explicit to implicit knowledge, thus closing the circle.

Corporate knowledge management is a complex process which involves human interaction and networking, resources, and technology that contributes to a solid and effective infrastructure. From a technological viewpoint, transformation and maintenance of knowledge can be supported by various tools. For example, groupware tools support teams and communities in sharing their knowledge (without necessarily having to make it explicit), computer-aided learning helps to internalize knowledge, and document management systems help to organize

and access documented knowledge. A more detailed discussion of transformation processes and their support by different technology bundles can be found in [BöKr99].

Market analysts regard the field of managing and analyzing unstructured information (such as document collections) as an important trend in supporting knowledge management¹. This is due to the fact that corporate knowledge can be contained in many of a company's documents. For example, consider engineering-intensive organizations like the chemical industries: Here, management documents, requirement definitions, technical guidelines or manuals of chemical plants contain important information about the company's goals, issues regarding configuration and maintenance of machines, or technological developments. An important element in the mosaic of knowledge identification and combination is to obtain a structured overview of such documents, thus enabling a deeper analysis and understanding of inherent text relationships and knowledge patterns.

As a real-world example consider patent analysis at British Telecom (BT) [WaDa00]: The company is particularly interested in how its patent portfolio compares with that of competitors. Questions which arise concern the areas of competence, the differences in emphasis or the technological trends in the market. Therefore, analysts have originally examined patent abstracts in a time-consuming and tedious manual procedure. BT's computer-aided solution for this problem was to adopt an automatic text classification approach: Based on keywords provided by the analysts the classification system computes a list of closely matching patent abstracts for each technology area. The quality of classifications depends, of course, on the suitability of provided keywords. Furthermore, relationships between abstracts within the often large classes remain unclear – and the long result list of the system still have to be examined manually. Another example can be found in the field of requirements engineering: Scenarios describe processes and interactions related to a system under examination, often in a textual format. Handling collections of weakly structured textual scenario descriptions is still a problem (cf. [JTC98]). Especially in large-scale distributed projects where many different developers produce a considerable collection of scenarios, the textual descriptions have to be compared, synchronized and categorized, making it necessary to work out the relationships of individual documents [JBK+99]. Manually managing this collaboratively elicited knowledge is a tedious procedure which could significantly benefit from adequate support.

Thus, finding and studying different techniques which effectively support the exploration and analysis of document bases is an important research issue. Beyond the use of document management systems, which mainly focus on filing documents and supporting documentation workflow, the need for accessing and structuring document collections based on their contents dramatically arises nowadays. Information retrieval systems aim at enabling a content-based and query-driven (goal-directed) search in document collections. While in this case the user has a pretty good understanding of what he is looking for, the need for structure and semantic “search paths” for a document collection arises when it comes to exploring document bases. The investigation of information spaces (like document collections) in an explorative way is a rather new field that has been titled information access [Stü95]. It complements the classic information retrieval paradigm. Typical tasks that are important in this context include searching for trends, comparing and aggregating information, detecting connections and links in data (cf. [Hea99]), or locating, categorizing and clustering items, comparing entities and relations, and associating and correlating objects (cf. [WeLe90]).

¹ See for example “Das Wissensmanagement organisiert den internen Informationsfluß effizienter”, Computer Zeitung Nr. 47/19.11.1998

Summing up these practical observations and putting the theoretical discussion into concrete terms, there are many aspects of working with documented knowledge that are valuable to support in practice. Relevant analysis tasks include pattern discovery of some kind, getting an overview of the topical document distribution and density of a collection, finding overlapping topic areas, examining content-based relationships between individual documents, or identifying key documents and unique texts within topic fields. Companies usually have a specific area of competence and interest and can predefine meaningful subsets of documents for a detailed analysis. Thus, specialized and quantitatively rather limited collections are of particular interest. This thesis tackles the sketched problem of effectively supporting the exploration and fine-granular analysis of specialized corporate document collections.

1.2 Solution Idea: Providing Visual Access to a Text Corpus

Text mining has recently started to emerge as a rapidly growing research field which focuses on the analysis of corporate document collections and the extraction of useful knowledge from text. Important aspects of text mining or knowledge discovery from textual databases include information and theme extraction, personalized content filtering, use of domain knowledge, text categorization, and – last but not least – document clustering and visualization. Graphically displaying complex information directly appeals to the powerful human visual perception which enables the user to rapidly identify patterns, trends and anomalies in large amounts of data. Visualizing the structures of document collections seems to be a straightforward consequence: For the analysis of a corpus of documents, visualization promises a means of easily identifying outliers, associations and clusters in the document space, thus pointing out the structure of corporate document collections. However, experiences with graphical overviews for information retrieval are rather disappointing (cf. [Hea99]): Up to now, research has failed to prove the usefulness of graphical collection overviews for the information search process. Collection analysis for knowledge management purposes, on the other hand, is a hardly considered application field. For exploration and pattern discovery in text collections structural overviews seem to be more important than for goal-directed search processes and promise to be helpful. This is the hypothesis studied in this thesis.

From the perspective of information visualization, “*a key research problem is [...] to discover new visual metaphors for representing information and to understand what analysis task they support*” [GeEi98]. This work proposes to visualize a document collection’s structure by means of document maps in order to support corpus analysis in knowledge management. *Document maps* present the overall similarity structure of a corpus of texts by using a suitable visualization metaphor which is reminiscent of geographical or astronomical cartography. More specifically, a document map technique is proposed which visualizes the topological structure of a collection of texts, thus supporting the analyst in discovering patterns in the document space and getting an overview of available material and its inherent relationships.

In this context some important technological aspects have to be considered: First, the quality of the visualized relationships crucially depends on the document analysis method. Even brilliant visualization methods cannot improve the quality of this component. Consequently, the proposed method is designed in a strictly modular way, such that collection-tailored document analysis methods can be applied, thus providing a reliable basis for displaying the structure of specialized document collections in the context of knowledge management. The second aspect is that visualizing thousands and millions of documents can only provide a coarse overview of the corpus – if at all. This work focuses on moderately-sized text collections. For a fine-granular understanding of the structure and relationships of documents within a special-

ized corpus individual documents should be visible and the grouping of related single documents and the grouping of text clusters should be powerfully visualized. The proposed document map approach therefore uses the visualization method of the data mining tool MIDAS [Skl96, SBJ99] which proved successful for knowledge discovery in structured data sets. This method is optimized for displaying cluster structures of small and medium data sets in a fine-granular way, thus enabling users to study detailed relationships between individual objects. The third aspect is concerned with incorporating an individual ‘bias’ of the analyst into the document map generation process: Since different analyst may have different personal weightings of document relationships they are interested in, the document map should offer a means for personal adaptability. Therefore, an extension of the proposed approach provides a semantic refinement component which allows the analyst to influence the structures presented in the visual display by incorporating scenario-related background knowledge and by expressing personal weightings of relationships.

The proposed document map approach is adaptable in two ways: It is (a) adaptable to the application domain by the exchangeability of the document analysis method, and it is (b) adaptable to the personal interest of an analyst by the semantic refinement component. In the remainder of this thesis ‘adaptability’ refers only to the second (more dynamical) meaning in order to avoid confusion. Altogether, the sketched solution idea of providing visual access to a corpus of corporate documents promises to support analysts in exploiting and fruitfully handling valuable documented knowledge.

1.3 Research Methods and Contributions

This thesis studies the use of graphically presenting the similarity structure of a text corpus for supporting analysis tasks in the context of managing the knowledge contained in specialized document collections. As a starting point a survey in knowledge-intensive companies and a study of document-intensive scientific and industrial projects yields an overview of analysis tasks which are important in practice. Condensed in a domain-tailored task model this overview is used for guiding an application-oriented design and a task-oriented evaluation of a document map approach for visually aiding the management of documented knowledge.

Regarding the research methods used, this work consists of two parts: The *conceptual and technical part* mainly uses engineering methods in order to analyze, adapt and combine existing methods and to design and adapt algorithms that bridge the gaps between them. For example, the proposed approach for generating document maps composes well-known methods from explorative data analysis in a modular design so that collection-tailored and application-specific document analysis methods can be incorporated. The semantic refinement component extends this basic design: Specifically developed algorithms connect this component to our framework, and matching methods for user-defined rules are developed based on results from information extraction and shallow natural language processing. A profound evaluation of the approach in the intended application domain is just as important as the developed method and the interactive document map system which is based on it: The hypothesis that document maps can successfully support corpus analysis in knowledge management requires a careful investigation of its use and value in real-world contexts and for real-world tasks. Consequently, the *evaluation part* of this thesis uses different empirical research methods which complement one another: Technical experiments with some test collections evaluate the technological quality and scalability of the approach. Case studies intensively examine the approach in real-world contexts and collect information on its usefulness on a qualitative level. Besides, this part of the evaluation effort yields insights in how to productively apply the

method in the application context. Furthermore, early case studies could contribute to an improvement of the overall technical approach. The task model for text corpus analysis in knowledge management is used to achieve more general and quantitative results on the effectiveness of supporting corpus analysis by document maps: In a controlled experiment the concept of visualizing a text collection's semantic structure is isolated and evaluated in a task-oriented way.

The contributions² of this thesis can be considered from two complementary points of view: From the perspective of the *application field of knowledge management* it

- provides an *advanced tool support* for the important and difficult problem of exploiting documented knowledge,
- shows meaningful and useful *application scenarios* for document maps and proposes methods for their *effective use* in these contexts, and
- presents *experience reports* from which other real-world projects concerned with structuring, condensing and exploiting documented knowledge can benefit.

From the perspective of *information visualization* the thesis

- *studies a visual metaphor* for a clearly defined application domain and contributes to the understanding of the analysis tasks this metaphor supports, and
- *shows the application potential* and the limits, the strengths and weaknesses of the visualization method and thus may help to improve existing and to develop new approaches for graphically displaying complex textual information.

1.4 Thesis Outline

Logically, this thesis consists of four parts: The first part presents a detailed analysis of the problem. Part two examines the state-of-the-art of document indexing and comparison methods as well as techniques for structuring and visualizing document collections. As the major result, motivation, goals and requirements for the own developed document map method are worked out. The third part contains the methodical and technical results of the thesis: A document map approach which is designed for supporting corpus analysis tasks in knowledge management and its realization in an interactive map-centered system is presented. This part of the thesis is further subdivided: Whereas chapters 6 and 7 present a basic document map approach which is thoroughly evaluated in the context of the intended application domain in part four, chapter 10 presents an extension of the basic method along with first experiments for its evaluation. The next sections describe the structure of the thesis in more detail.

Problem Analysis

Chapter 2 presents a task model which describes typical facets of practical text corpus analysis tasks. It provides a taxonomy for characterizing, comparing and classifying tasks that are relevant in knowledge management and can potentially be aided by visual text-access technology. The taxonomy is based on selected task models from literature, analysis tasks examined in document-intensive scientific and in-

² A more detailed list of contributions will be presented in section 5.8.5 after a deeper problem analysis and a description of the technical state-of-the-art.

dustrial projects, and findings of a survey in knowledge-intensive industries. The resulting model is used for analyzing the requirements of the own technical approach. Moreover, it yields the basis for an intensive, task-oriented evaluation.

State-of-the-art in Structuring and Visualizing Text Corpora

- Chapter 3 presents document representation and comparison methods which have been proposed in the fields of information retrieval, information extraction and case-based reasoning. The aim is to examine the spectrum of techniques available for analyzing document collections in different, possibly specialized application fields.
- Chapter 4 sketches a selection of methods for visualizing the similarity of data objects in the application field of exploratory data analysis. These techniques are assumed as known in the technical parts of this thesis.
- Chapter 5 reviews methods for structuring and visualizing document collections. The focus is on methods which visualize the similarity of documents within a text corpus; other work has certain links to the problem of computing and visualizing semantic text relationships. The approaches are compared in the context of needs for visualizing fine-granular relationships of textual documents in the application area of managing the knowledge contained in specialized document collections. Based on the state-of-the-art and the intended application domain, the necessity of further research is indicated, and motivation and goals of the own approach are summarized.

Presentation of the Developed Method and its Implementation

- Chapter 6 introduces a modular framework for generating document maps. In particular, the framework's design is motivated, its modules and interfaces are formally specified, and concrete methods for its realization are adapted and presented. Finally, the interplay among the modules is discussed and hints for setting the framework's parameters are given.
- Chapter 7 presents the prototypical realization of the approach proposed in chapter 6 and its incorporation into an interactive system environment. Objectives for developing the system are discussed, a suitable system architecture is developed and its connection to the modular framework for generating document maps is sketched. The graphical user interface of the system is presented and additional interactive features along with the methods behind them are introduced.

Evaluation

- Chapter 8 contributes to a deeper understanding of the usefulness of visualizing the semantic structure of corporate text collections. It presents the application of the document map approach to practical corpus analysis tasks and a qualitative evaluation of the support offered by visually displaying the similarity structure of text corpora. Three case studies in scientific and industrial environments are presented.
- Chapter 9 presents a comparative study in which the basic concept of the document map approach – the visualization of the semantic structure of a text corpus – is evaluated in a task-oriented laboratory study.

Extension

Chapter 10 deals with the question of how to incorporate adaptability into the basic framework, so that analysts can influence a document map regarding their personal interest. A suitable methodical and technical approach is proposed and first evaluations in experimental settings are reported.

2 A Task Model for Text Corpus Analysis in Knowledge Management

„Just as people have mental models of information systems, many information systems attempt to build models of the people using them.“
[AlMi90]

In the present information intensive era knowledge constitutes a valuable resource for many enterprises. Since a great deal of strategically relevant information is ‘stored’ in information sources like collections of unstructured text documents, suitable text-access mechanisms play an important role. This chapter examines the types of tasks that are relevant when it comes to accessing and analyzing documented knowledge sources. More precisely, the focus is on analysis efforts where the user is interested in gaining insight into a text collection’s structure as a whole and in analyzing aspects of the text corpus, e.g. figuring out relationships of documents. By abstracting from concrete tasks a general task model for text corpus analysis in knowledge management is set up, i.e. a taxonomy which supports the description of important aspects of tasks which are relevant in analyzing document collections. The task model shall help to typify analysis tasks by their dominant features, so that they can be generalized and compared.

More generally, task models are user-centered representations of goals and actions a user needs to perform in the context of information processing (cf. [MLO00]). They are important for a structured view of the types and scopes of user actions in certain application domains. Such a model provides a suitable taxonomy for describing tasks which might be effectively supported by current or future systems. There are many task models in the context of working with visual information retrieval systems (some of them are presented in this chapter). So far, however, there is no model which is specialized for document corpus analysis tasks that are relevant in knowledge management and which can potentially be aided by visual text-access technology.

Such a model could be applied for various purposes, such as supporting requirement analysis in system engineering where an important first step towards application-adequate system modeling is to understand actual users needs. Another application from a more user-focused view is studying and assessing task types which arise in the considered domain. Given that the model provides a precise, system-independent taxonomy it can be used for describing, classifying and evaluating information systems regarding to the user-oriented tasks they support (cf. [MLO00]). Since interaction is a crucial but complex factor for modern information systems, a task model for corpus analysis tasks can make a contribution to improve visual user interfaces. In the context of this thesis the resulting model contributes to a deeper understanding of the application domain under examination, helps to analyze the technical requirements, and can be used as a yard stick for a task-oriented evaluating the own approach.

The task model developed in this chapter is based on three sources: Selected models from literature (sections 2.2 and 2.3) are condensed to form a theoretical basis for the development

of a specialized taxonomy (section 2.4). A practical investigation of text corpus analysis missions in scientific and industrial projects provides well-understood real-world tasks, thus yielding an important source of knowledge for reviewing and refining known models (section 2.5). Findings of a survey in knowledge-intensive industries complete and optimize the specialized task model for corpus analysis in knowledge management (section 2.6). Details of this study have been worked out in [Seel01]. Parts of this chapter have been published in [BeSe01].

2.1 General Definitions and Requirements

In this work a *task* is defined as a complex formulation of a problem which is characterized as follows:

- there is a *goal* which can be described at least in an abstract way,
- several physical or cognitive *actions* are necessary to reach the goal,
- in general there are different *ways and strategies* to solve the problem.

A task is regarded to be *important* if it is

- *typical*, i.e. it occurs frequently in the context of knowledge management,
- *difficult* to perform and thus can benefit from suitable support,
- *valuable* for the user to solve it, and
- *information technology* can possibly support the problem solving process.

Requirements for the desired task model include:

- *Accuracy of terminology* is a basic requirement since precise and clear task descriptions are desired.
- The model shall be *flexible* in the sense that the degree of abstraction of task descriptions should be variable because in practice tasks with different complexity have to be characterized and compared against each other. It thus should be possible to find different, yet not trivial levels of abstraction.
- It is desired to have a *domain-specific* model which is optimized for characterizing analysis tasks of a document corpus in knowledge management. Furthermore, the model has to be suitable for evaluating visual interfaces which aim at supporting the access of document collections. It thus should comprise, for example, a means to describe the considered document relationships for certain tasks.
- The model should be *extensive* in the sense that important analysis tasks from document analysis in knowledge management can be characterized and compared adequately using the provided taxonomy.
- The model should be *extendable* so that necessary adaptations can easily be performed.
- Last but not least, the model has to be *plausible*. Precision and applicability to the considered domain are regarded to be more important than theoretical completeness.

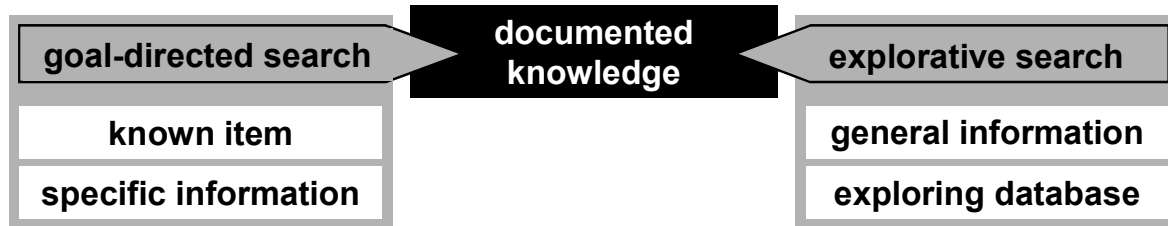


Figure 2-1: Accessing documented knowledge: goal-directed versus explorative search

2.2 Goal-Directed versus Explorative Document Access

A first step in setting up a taxonomy for the description of corpus analysis tasks is to look at the terminology for document search in information retrieval – a research field which is concerned with searching for information in unrestricted document collections. Document search has different facets, depending on the specific kind of information need which motivates it. An information need arises when a specific set of knowledge is necessary to solve a certain task or to answer a certain question. The need implicitly defines a goal for the search. Meadow [Mead92] distinguishes four different types of search:

- In a *known-item search* the source of the desired information is already known and can be described in detail, but its physical location is unknown. For example, consider a user looking for the library signature of a book he wants to borrow or an employee trying to retrieve a concrete enterprise document. This is the most specific kind of search as the goal can be stated precisely.
- A *specific-information search* aims at finding documents regarding a certain well-defined question or task but no concrete item containing the information is known. The goal of the search is quite clearly specified and can be formulated sufficiently. At least suitable attribute values are known to initialize the search with. For instance, think of an engineer who seeks information about properties of melt flow in steel casting. However, the formulation of the query may change over time, influenced by the material found during the session.
- A *general-information search* is characterized by a broader information need which is hard to formulate. The search goal comprises various topics which can be subsumed by a generic term, thus addressing multiple targets. Consider a knowledge manager who tries to figure out what is known about steel casting technology in his enterprise in general. Relevant material cannot be described *a priori* but depends on the contents of the database. The searcher may have to trace the items within the database to identify interesting pieces of information (“I know it when I see it”), albeit he or she may still miss relevant documents.
- *Searching to explore the database* aims at getting familiar with the contents of a document base. This kind of search is driven by a higher interest, e.g. a certain analysis task, rather than motivated by a merely specified goal. For example, in a system analysis effort the consultants may want to find out what kind of knowledge is stored, what business processes are reflected in the produced documents, etc. In fact, the user is looking for meta-information, patterns or trends.

To summarize (cf. figure 2-1), the search types *known item* and *specific-information* are characterized by a sufficiently specified goal which can be expressed in terms of a query and thus direct the search. These search types can be subsumed by the term *goal-directed search*. Searches of the types *general information* and *exploring the database* are motivated by a more general information need. Their goals do not provide sufficient clues to lead the search and can be reached merely by interacting with the database. Thus, the process of searching depends on the collection itself and therefore can be called *explorative*. The most prominent explorative search strategy is browsing – an interactive process in which a large amount of information is scanned and a mental model of the information space is built. Note that some authors use the term ‘search’ when they refer to goal-directed corpus access and ‘browsing’ or ‘scanning’ for explorative access.

2.3 A Brief Review of Task Models

A prominent approach to model tasks originates from the KADS methodology (Knowledge Analysis and Documentation System, [WSB92]) for modeling knowledge. This framework includes the concept of task models which describe tasks that are relevant in a particular problem solving process in a detailed way. The relatively complex KADS task models decompose high-level tasks of application domains into more specific sub-tasks. Generic task models are partially instantiated skeleton models for typical tasks or task fragments which include data-flow (inputs and outputs required by primitive tasks) and control structures. In contrast to such a detailed description of the course of task actions in a particular application domain, the focus of task models for (visual) information retrieval is on describing the characteristics of typical text access tasks on a general level. This section reviews several of these task models. The aim is to find a suitable basis for the desired domain-specific taxonomy. A good overview of task models for text access can be found in [Mor99].

2.3.1 Overview and Classification of Prominent Task Models

Task models can be roughly classified according to their degree of granularity and their dependence or independence of a domain (cf. [Mor99]). *Granularity* refers to the nature and complexity of modeled tasks. The spectrum reaches from single actions performed by a user or system up to abstract user objectives. In the following we refer to elementary tasks as ‘units of the model’. A unit can be a system action, a user action or a feature of a task. If single user actions are the units of a task model, cognitive and external (physical) actions [SuEn96] can be distinguished. Since there are varying definitions of the notion of ‘task’ it is sometimes hard to compare different models.

A task model can be *domain-dependent* or *domain-independent*. Domain-dependent models can be further classified regarding to their degree of domain-dependence (e.g. task models for text retrieval in general or models for retrieving bibliographic data). Domain-independent models are generalized descriptions of tasks and thus usually of a coarse granularity. However, they can be seen as meta-models from which domain-dependent models can be derived (a method which is pursued in this work). Another criterion of distinguishing task models is the modeled *type of relationship* between task units. Examples for possible relationships are logical sequences of actions or hierarchical relationships between tasks and sub-tasks.

Table 2-1: Units modeled by prominent task models for visual information retrieval

author	units modeled
Shneiderman 1996 [Shn96]	overview, zoom, filter, details on demand, relate, history, extract
Wehrend & Lewis 1990 [WeLe90] presented in [MLO00]	locate, identify, distinguish, categorize, cluster, distribution, rank, compare within entities, compare between relations, associate, correlate
Zhou & Feiner 1998 [ZhFe98]	visual grouping, visual attention, visual sequence, visual composition, structuring, encoding, modification, transition; (elementary actions as in the model of Wehrend & Lewis)
Marchionini 1992 [Mar92] presented in [MLO00]	define the problem, select the source, articulate the problem, examine the results, extract information
Bates 1989 [Bat89] presented in [MLO00]	footnote chasing, citation searching, journal run, area scanning, subject search in bibliographies and abstracting and indexing service, author searching
Belkin et al. 1994 [BST94]	method of interaction (scan vs. search), goal of interaction (learn vs. select), mode of retrieval (recognize vs. specify), resource considered (information vs. meta-information)

Table 2-2: Characterization of prominent task models for visual information retrieval

model	modeled units	# units	domain	type of model
Shneiderman	system functions	7	domain-independent	set of units
Wehrend & Lewis	user actions	11	domain-independent	set of units
Zhou & Feiner	user actions	50	domain-independent	type hierarchy
Marchionini	user actions	5	retrieval and browsing	sequence of actions
Bates	user actions	6	literature retrieval	set of units
Belkin et al.	task features	4	domain-independent	feature space

Table 2-1 briefly presents the units of prominent task models from literature, table 2-2 characterizes the models with respect to the classification criteria sketched above. Morse [Mor99] identifies different weaknesses of the presented models with respect to the demand of having a user-centered design approach to defining tasks: They are either system-oriented instead of user-oriented, too general or too specialized, or weakly structured. Regarding their suitability as a basis for the desired domain-dependent task model for corpus analysis two more drawbacks arise: The models are incomplete since they neglect important features which differentiate tasks, and some possible combinations of criteria seem to be irrelevant in practice, thus leading to a blurred picture of reality.

Which task models may be of use for setting up a new, domain-dependent model? Obviously, the Marchionini approach is strongly related to goal-directed, query-driven processes instead of allowing the description of analysis tasks. It thus will be excluded from further considerations. The model of Bates is not suitable either since it is designed for describing literature searches. The task model of Zhou & Feiner is an extension of Wehrend's and Lewis' approach. Both models contain the same elementary actions which can be combined to more complex tasks and can be adapted to the application domain of corpus analysis in knowledge management. Due to its lower complexity only the model of Wehrend & Lewis will be considered in the following. The remaining models shall be studied more intensively. The following presentations are guided by the model reviews in [Mor99] and [MLO00].

2.3.2 Detailed Presentation of Selected Task Models

2.3.2.1 Wehrend & Lewis

The approach of Wehrend & Lewis [WeLe90] aims at classifying functions of visual information retrieval systems. It distinguishes 11 tasks that a user can perform in such a system. The following list presents the modeled set of actions:

- *Locate*: Includes all techniques which allow the user to find special objects, e.g. highlighting objects or areas in a visual presentation or searching known items.
- *Identify*: This function type can be used to specify and search possibly unknown objects.
- *Distinguish*: Functions for visually distinguishing different values of a variable (e.g. relevant – not relevant).
- *Categorize*: Comprises actions for grouping objects with respect to certain criteria.
- *Cluster*: Techniques for displaying the structure of the object space, e.g. mapping objects from high-dimensional feature spaces into 1D, 2D or 3D spaces (user action: examine structure of object space).
- *Distribution*: This action is similar to cluster, but here a pattern description has to be specified (which enables users to examine the structure of the object space according to given criteria).
- *Rank*: Functions of this type allow a sorted presentation of scalar or ordinal data (user action: sort objects according to given criteria).
- *Compare within entities*: Actions of this type describe tasks in which a user specifies attributes of similar objects and compares the entities according to their features.
- *Compare between relations*: Techniques for comparing different objects or groups of objects regarding common features or differences.
- *Associate*: This action asks the user to form relationships between displayed objects.
- *Correlate*: Techniques for grouping objects according to any of their attributes, e.g. arranging symbols of text documents according to their length or their author. The user can work out similarity relationships based on specific object features.

Though the model does not explicitly formulate relationships or dependencies of the different types of actions they can be given additionally. The extension by Zhou & Feiner [ZhFe98] allows to group actions to more complex tasks.

2.3.2.2 Shneiderman

The model of Shneiderman [Shn96] (probably the most prominent approach) is a framework for comparing existing systems for visual information presentation and for supporting the design of new advanced visual user interfaces. It offers a task by data type taxonomy with seven data types (1-, 2-, 3-dimensional data, temporal data, multi-dimensional data, tree and network data) and seven tasks which correspond to the interface functionality. Since the focus

of this chapter is on characterizing tasks, not systems, only the interface functionality is considered. Similar to the taxonomy of Wehrend & Lewis this model is rather system-oriented, but the described actions can also be interpreted as user actions:

- *Overview*: Techniques that display the entire space of objects (user action: gain an overview of the entire collection).
- *Zoom*: Tools for restricting the view to specified items of interest (user action: set focus to an interesting subset of objects for further actions).
- *Filtering*: Techniques for excluding uninteresting items (interpretation as user action similar to ‘zoom’).
- *Details-On-Demand*: Functionality to get details of an item or group when needed (user action: fix interesting objects and closely examine their properties).
- *Relate*: Techniques for displaying relationships of items in the object space (user action: examine relationships among objects).
- *History*: Functions which allow the user to undo, replay and progressively refine actions (user action: keep track of complex exploration tasks and re-use experience from past actions).
- *Extract*: Functions which allow the extraction of sub-collections and of query parameters in order to save it to a permanent data memory.

2.3.2.3 Belkin et al.

The model of Belkin et al. [BST94] aims at characterizing general searches in information retrieval. It comprises four two-valued criteria and thus allows to distinguish 16 different search processes.

- *Method of interaction (scan vs. search)*: The notion of ‘search’ refers to finding known items or items which can be described specifically. In contrast, ‘scan’ means ‘trying to find something interesting’ without necessarily knowing or being able to describe what exactly is wanted.
- *Goal of interaction (learn vs. select)*: This criterion distinguishes the goal of learning something about the considered objects or their relationships and the goal of actually retrieving items.
- *Mode of retrieval (recognition vs. specification)*: Objects to be found may be specified (the formulation of an information need is dominant) or simply recognized (‘I know it when I see it’).
- *Resource considered (information vs. meta-information)*: Is the user interested in information inherent in the searched object or in information about an object?

2.4 A New Basic Task Model Derived from Literature

The task models of Shneiderman, Wehrend & Lewis and Belkin et al. will be used as a basis for a new, specialized task model for the analysis of document collections in knowledge management as they are all domain-independent models for visual retrieval and analysis tasks.

2.4.1 Comparison of Selected Task Models

Comparing different task models is a difficult matter since it is necessary to deal with differing notions: User actions, system actions and criteria of tasks have to be compared. However, a combination of the models is possible on a more general level of abstraction since user actions and system functions are closely connected due to the concept of interaction (e.g., Shneiderman's tasks have already been interpreted as user actions in the model's presentation).

Thus, the first thing to do is to choose a suitable reference model within which the taxonomies of the other models can be interpreted. The Shneiderman approach is rather system-oriented than user-centered. Only the unit 'relate' reflects cognitive user actions (cf. [Mor99]). Thus, for developing a system-independent task model this approach is no suitable reference taxonomy, though it implies interesting user actions which can be used to enrich other models. The defined units in the taxonomy of Wehrend & Lewis are too fine-granular to serve as a reference model. In contrast, the criteria of the Belkin approach set up a four-dimensional space which is general enough to include the task types of the remaining models, thus enabling to relate the actions of the other taxonomies to Belkin's criteria. This kind of comparison will yield a better understanding of differences and common grounds.

To begin with the construction of a new basic model, it is necessary to find out if and how system or user actions of the other models can be characterized by Belkin's criteria. Having found such characterizations the model of Belkin can be refined by considering the respective criteria of Wehrend & Lewis and Shneiderman. Note that the criteria of Belkin originally are meant to be of a general nature. Now, however, they may be related to fine-granular actions which forces a new interpretation. For example, the criterion 'goal of interaction' which originally describes the goal of a complex process, is now to be understood as the goal of a single action.

Table 2-3 compares the approaches of Shneiderman and Wehrend & Lewis with Belkin's model. Each action of the respective model is characterized by a criterion-value of Belkin. An empty field means that the unit of the considered model cannot be characterized by (only) one of the Belkin criteria-values. Since this comparison requires an interpretation of inherently vague criteria and values there might be alternative assignments. However, the table presents a plausible interpretation and can serve as the starting point for setting up a new model derived from theoretically grounded approaches.

In order to find out which criteria of Belkin could be enriched or refined, the table can be analyzed according to the following aspects:

- In a horizontal analysis of table 2-3 combinations of Belkin values which *do not occur* can be figured out. It should then be examined if such a combination would be meaningful at all.
- In a vertical analysis of table 2-3 Belkin values which *occur frequently* can be examined. It should be checked whether new values can be added to better distinguish tasks.

Table 2-3: Comparing Belkin's model against the task models of Shneiderman and Wehrend & Lewis

		Belkin et al.			
		method of interaction: scan vs. search	goal of interaction: learn vs. select	mode of retrieval: recognize vs. specify	resource considered: information vs. meta-information
Shneiderman	overview	scan	–	recognize	meta-information
	zoom	scan	learn	recognize	meta-information
	filter	–	–	–	meta-information
	details on demand	–	learn	recognize	information
	relate	scan	–	recognize	meta-information
	history	–	learn	recognize	meta-information
	extract	–	select	–	–
Wehrend & Lewis	locate	search	select	specify	information
	identify	search	learn	specify	information
	distinguish	scan	–	recognize	meta-information
	categorize	–	select	specify	meta-information
	cluster	–	learn	recognize	meta-information
	distribute	–	learn	recognize	meta-information
	rank	–	–	specify	meta-information
	compare within	scan	learn	recognize	meta-information
	compare between	scan	learn	recognize	meta-information
	associate	scan	–	specify	meta-information
	correlate	–	select	–	meta-information

- Concerning actions which cannot be described adequately by Belkin values (*empty fields* correspond to situations where no value or both values would be adequate to characterize the action) it can be examined whether a new criterion should be added in order to be able to characterize tasks more detailed or whether an existing criterion can be enhanced by adding a new value.
- Finally, entire *rows* of the table can be compared: Which actions cannot or only poorly be distinguished by value-combination of Belkin's model?

As an example of the horizontal analysis consider the values 'recognize' and 'search' which do not occur in combination. This observation reveals a dependency between criteria of the Belkin model: When known items or specific objects are searched, specifying is usually more important than recognizing the result. Regarding the vertical analysis it turns out, e.g., that the value 'meta-information' is very general and should thus be subdivided into finer values. Table 2-4 summarizes the results of the vertical analysis and presents a characterization of Belkin's criteria values by actions of the other models. Considering the empty fields in table 2-3 it can be noticed that the model of Belkin forces indifferent characterizations of tasks related to grouping objects (e.g. 'categorize', 'cluster', 'distinguish') and of tasks that comprise the examination of objects regarding similarity relationships ('correlate', 'rank'). Finally, some important actions of the other models can only poorly be distinguished or even fall together in Belkin's taxonomy (e.g. 'compare within entities' and 'compare between relations'). It is thus necessary to perform some refinements by introducing new characterizing values and criteria. This is particularly necessary in order to put the model in the concrete form of a

Table 2-4: Actions related to criteria values of Belkin's model

method of interaction		goal of interaction		mode of retrieval		resource considered	
scan	search	learn	select	recognize	specify	information	meta-inf.
overview, zoom, relate, distinguish, compare within, com- pare between, associate	locate, identify	zoom, details, history, iden- tify, cluster, distribute, compare within, com- pare between	locate, catego- rize, correlate, extract	overview, zoom, details, relate, history, distinguish, cluster, dis- tribute, com- pare within, compare between	locate, iden- tify, catego- rize, rank, associate	details, locate, identify	overview, zoom, filter, relate, history, distinguish, categorize, cluster, dis- tribute, rank, compare within, com- pare between, associate, correlate

corpus analysis task model. A more detailed discussion of observations would be beyond the scope of this section.

2.4.2 A New Basic Task Model

Belkin proposes the 16 combinations of the four two-valued criteria for describing tasks. However, the model is too abstract to adequately describe useful actions from other models. Of course it is desirable to have as few types of tasks (or criteria combinations) as possible but it is also important to allow as many combinations as necessary: There are analysis tasks and actions which could be characterized by both values of a criterion. A forced decision for one value would only apparently be precise.

As a consequence, the *structure of the model* is changed from a feature space to a hierarchy of criteria-values. First, the new model's criteria are multi-valued. Since a task may be complex, any combination of values for each criterion is allowed. This allows to express that in a certain task some values of a criterion may be valid at the same time. Though this makes the model more complex, the precision of task characterizations will profit from that. For example, consider the task 'getting familiar with a collection of documents'. Typically, this task involves scanning the collection in order to develop hypotheses of its contents, and searching the collection in order to check these hypotheses. Thus, learning about the collection's contents is an alternating process of scan and search. The final structural change is that the values for criteria are organized hierarchically, i.e. each criterion may contain values which can be considered on different levels of granularity. This allows to compare tasks on different levels of abstraction which is especially useful when tasks of varying complexity are considered.

Regarding *aspects of contents*, first of all the abstract notions of 'objects' or 'items' are domain-specifically replaced by 'documents' (which, of course, has effects on the interpretation of actions). As an additional goal of analysis, besides learn and select, 'classification' is introduced since tasks which aim at distinguishing groups of documents differ in some respects from other tasks of retrieval and analysis. Concerning the aspect 'resource considered', the notion of 'meta-information' is very general and shall thus be determined more precisely: On the one hand, document attributes (like 'author' or a pre-defined class) can be an important information about documents, on the other hand the similarity structure of the corpus may be considered. For example, when it comes to inspecting a given classification the considered resource may be document attributes (in this case class information attached to each document). When a classification shall be defined, in contrast, the analyst may be more interested in structural information (e.g. the cluster structure of a collection). This differentiation is simi-

Table 2-5: Basic task model derived from literature

goal of interaction	learn	
	select documents	
	classify	
resource considered	information from documents	
	meta-information	document attributes
		structural information
method of interaction	explorative (scan)	
	goal-directed (search)	
focused relationship	document – document	
	document – topic	
	topic – topic	
mode of retrieval	recognize	
	specify	

lar to the action types ‘compare within entities’ and ‘compare between relations’ in the task model of Wehrend & Lewis (cf. section 2.3.2.1). Regarding the aspect ‘method of interaction’, the term ‘scan’ will be replaced by ‘explorative search’ and ‘goal-directed search’ will take the place of ‘search’, following 2.2. Finally, an additional criterion – called focused relationship – is introduced in order to be able to adequately describe tasks in which the comparison of documents or document groups (‘associate’, ‘correlate’, etc.) plays an important role. The resulting basic task model is sketched in table 2-5. A detailed description of the model will be given at the end of this chapter after having performed an optimization.

2.5 Enriching the Basic Model: Analysis of Real-World Tasks

The preliminary task model for corpus analysis in knowledge management as constructed in the last section is based upon theoretical work from the field of (visual) information retrieval. Some basic requirements (cf. section 2.1) are that the resulting model is domain-specific, extensive and plausible in the considered application domain. It is thus necessary to examine and optimize the basic model with respect to real-world analysis tasks.

2.5.1 Extracting Corpus Analysis Tasks from Practical Applications

In a first step typical real-world analysis tasks were listed. These tasks originate from two sources: On the one hand, well-understood analysis actions were collected by an investigation of text corpus analysis missions in some large-scale scientific and industrial projects. These case studies will be presented in chapter 8 in detail in the context of evaluating the own text-access approach. On the other hand, in business life of knowledge-intensive enterprises numerous tasks regarding document analysis arise which might be successfully supported by current or future information systems. In order to take these types of tasks into account and to broaden the basis for enriching the basic task model, a survey was performed in German companies.

The method of the survey shall be briefly sketched: Structured interviews (following a dedicated interview guide) were performed to collect data about corpus analysis tasks in knowledge management. Personal, structured interviews have some advantages over written questionnaires: They are more flexible and allow a deeper understanding of the raised data by the interviewer. Furthermore it is possible to adjust questions adequately to different interview-

ees. A reasonable degree of comparability is given by the underlying structure of the interviews. However, the effort of this kind of survey is high so that the possible number of interviews is strongly limited.

Questions asked in the structured interviews concerned the type and characteristics of documents, their value for the company, personnel who works with the documents, infrastructure for storing and accessing documents and, finally, typical tasks that arise for organizing, managing or exploiting document collections. The data raised by these questions were used to identify important, valuable and interesting analysis tasks for document collections which should be describable in the desired task model (cf. section 2.1). The interviews did not focus on standard retrieval actions (like known-item or specific information searches).

Due to restrictions in time and personnel capacity it was necessary to cover a broad field of tasks in as few interviews as possible. Consequently, companies originating from different knowledge-intensive fields were chosen and asked to participate in the survey. Altogether eight interviews (seven personal and one written, telephonically prepared interview) with experienced knowledge management personnel (such as documentation, archive and development manager) in the fields of software development, chemical and steel industries, newspaper archiving and library, and an R&D department of automobile industry were performed. It turned out that this apparently small basis is sufficient for our purposes since many tasks derived from the case studies and the first interviews occurred in similar form in later interviews. Thus, the task types ‘converged’ quickly and later interviews could consolidate the findings.

Based on these results, a list of tasks was compiled which comprises verbal task descriptions and explanations of their motivation along with a sketch of the types and characteristics of documents involved, and meta-information available for each task. Due to the heterogeneity of interviewed companies we could gain insight into the work with different types of documents and various underlying goals. Table 2-6 shows some typical sample tasks which frequently occur in similar forms.

Table 2-6: Sample tasks derived from the survey

task description	Check documentation regarding single sources and redundancy. Condense documents if necessary and set up a uniform information base.
motivation	Reducing costs of maintaining documentation, providing high-quality documentation bases.
problems of analysis	Heterogeneity of material, different authors produce similar material, consistency.
types of documents	Manuals, online documentation, tutorials, developer notes, specification documents, ...
meta-information	Possibly: types of documents.
task description	Check and refine a categorization scheme for text documents (e.g. in an enterprise documentation system).
motivation	Categorization schemes have to be adjusted for dynamic text bases. Unsuitable categories (too large, too small, too diffused) complicate archiving and retrieval.
problems of analysis	unstructured content of large categories, relationship among categories unclear
types of documents	articles, internal/external project documentation, correspondence, notes, ...
meta-information	given categorization scheme
task description	Develop a categorization scheme for text documents (e.g. for an enterprise documentation system).
motivation	Setting up a structured data store.
problems of analysis	missing overview of documents, mental effort due to large amount of data
types of documents	articles, internal/external project documentation, correspondence, notes, ...
meta-information	—

2.5.2 Optimizing the Basic Model

Having collected typical tasks which occur in the context of analyzing document collections in knowledge management, the second step is to optimize the basic model with respect to the clarity with which tasks can be described. This effort involves an iterative process of adjustment: The model has to be completed and refined for each task and, if necessary, pruned so that only highly descriptive and relevant criteria and values remain in the final model. More precisely, each iteration for adjusting the basic model contains the following steps:

1. Examine an analysis task: Give a general verbal description of the task and abstract from too specific notions.
2. Try to characterize the task with the task model taxonomy available so far.
3. Test if the task can be well-distinguished from other, intuitively different tasks.

If necessary, new values or new criteria can be incorporated into the model. In the end it is necessary to examine possible dependencies among criteria (cf. section 2.4.1): Value-combinations which co-occur frequently have to be discussed in order to decide whether they may depend on each other. If so, respective criteria or values should be combined or eliminated.

In order to illustrate the refinement process performed for the desired taxonomy consider the basic task model as presented in section 2.4 as well as the following task (cf. table 2-6):

Check documentation regarding single sources and redundancy. Condense documents if necessary and set up a uniform information base.

So far, the criterion ‘goal of interaction’ contains the values ‘learn’, ‘select’ and ‘classify’. The goal of the task is to assure the uniformity and quality of a collection of valuable documentation material. This, however, cannot be described by the existing values: Neither ‘learning’ nor ‘selecting’ (and certainly not ‘classifying’) is a sufficiently precise characterization of this action. Criteria for quality depend on the application and type of text. Since the models abstracts from concrete text types it seems sufficient to introduce a value ‘check documents’ for the criterion ‘goal of interaction’. During the further iterative refinement process it turned out that intuitively different tasks which share the broad goal ‘check documents’ need to be distinguished more carefully: ‘Controlling quality’ is a rather passive process in which the validity of quality criteria is checked, whereas ‘assuring quality’ is rather active in the sense that the analyst works with documents and collects information in order to influence them. Such a differentiation better reflects the different procedures applied.

Besides, the action ‘condensing documents’ plays an important role in many other contexts as well, thus making it useful to introduce a corresponding value for the ‘goal’ criterion. Furthermore, the raised task samples reflect that working with categories (categorization and classification) of different kinds (catalogues, file systems and similar) is significant and complex. As a consequence, a criterion ‘categories’ with corresponding values replaces the goal ‘classify’. Tasks with the goal of using or setting up categorization schemes can be interpreted as ‘quality control’ or ‘quality assurance’ tasks.

The pruning step in the end of the refinement process revealed and eliminated dependencies among criteria: Very early in the refinement process it seemed to be suitable to introduce a criterion ‘acting person’ with the values ‘management, maintenance personnel’ and ‘end user’. However, the values of this criterion clearly correlate with the (initially flat) values of the criterion ‘goal’ so that instead the goal values have been grouped into the more general values ‘making use of documents’ and ‘maintaining documents’.

Finally, all available task samples were described by the resulting taxonomy in order to validate the model, i.e. to check whether all tasks can be described with a sufficient degree of clearness. In particular: Does the model allow a valid grouping of similar tasks? Can important and sufficiently differing tasks be distinguished? The final model is described in detail in the remainder of this chapter.

2.6 The Resulting Task Model

The resulting task model is presented in table 2-7. It comprises eight criteria with their values and sub-values. Because accuracy of terminology is a basic requirement for the model the meaning of criteria and values shall be defined and contrasted. A specific interpretation, however, depends on the context of the analysis task actually considered. The usage of the model will be sketched in section 2.6.2.

2.6.1 Definition of Criteria and Values

This section defines the criteria and their values in more detail in order to disambiguate the notion. For each leaf of the criteria-value tree example tasks are given which could be characterized by the respective value.

GOAL OF INTERACTION: The goal characterizes the motivation for performing the task. It can be seen as a superordinated dimension of the analysis process. The super-value MAKING USE OF DOCUMENTS characterizes tasks in which analyzing a document collection is performed for

Table 2-7: Task model for analyzing document collections in knowledge management. Criteria and values marked by an asterisk (*) correspond to an item of Belkin's task model. The criterion 'mode of retrieval' is now named 'mode of communication' in order to differentiate it from a criterion for specific retrieval tasks.

goal of interaction *	making use of documents	learn *
		condense documents
		select documents *
	maintaining documents	control quality
		assure quality
dynamics of focus of interest	fixed	
	adaptive	
resource considered *	information from documents *	
	meta-information *	document attributes
		structural information
method of interaction *	explorative	
	goal-directed	
granularity	overview	
	details	
categories	not relevant	
	relevant	use existing
		check classes
		use classes
focused relationship	external: document – specification	
	inherent	document – document
		document – topic
		topic – topic
mode of communication *	recognize * (reflect, physically passive)	
	specify * (physically active)	

external (value-creating) purposes: The analyst may **LEARN** from or about documents. This is an internalization process of explicit, documented knowledge where information reception is emphasized. Or the analyst may **CONDENSE DOCUMENTS** in order to synthesize a sub-collection of documents (using the collection to generate new documents, e.g. summaries, extensions, etc.). This includes the association of documents. Third, **SELECT DOCUMENTS** describes tasks in which a set of documents (characterized by certain features) is identified and extracted from the collection. The general value **MAINTAINING DOCUMENTS** characterizes tasks in which the documents themselves are in the focus of interest in order to preserve their value: **CONTROL QUALITY** describes actions in which the validity of quality criteria is checked (a rather passive task without directly influencing the collection). **ASSURE QUALITY** is rather active in the sense that the analyst intends to work with documents and collects information in order to influence the corpus. Though these values are closely connected the differentiation better reflects the different procedures applied.

Examples:

- **LEARN:** Analyzing a collection of patents in order get an idea of a market.
- **CONDENSE DOCUMENTS:** Synthesizing project documents for an interim report.
- **SELECT DOCUMENTS:** For generating a project overview summary texts are searched.
- **CONTROL QUALITY:** Checking if documents are classified correctly.
- **ASSURE QUALITY:** Guaranteeing consistency of documentation.

DYNAMICS OF FOCUS OF INTEREST: This criterion characterizes the overall clarity of underlying search-goals. If the focus of interest is more or less **FIXED**, a more systematic work procedure is possible. On the other hand, the search-goal may be of coarse granularity, and the detailed need for information is **ADAPTIVE** with respect to the process itself: The focus is moved regarding information found during analysis.

Examples:

- **FIXED:** Finding documents which are related to a given document.
- **ADAPTIVE:** Getting an overview of existing material.

RESOURCE CONSIDERED refers to the source of information which is in the focus of interest. If knowledge is to be extracted from documents themselves the value **INFORMATION FROM DOCUMENTS** is appropriate. Knowledge may be derived from **META-INFORMATION** if the interest is comprehensive. In this case a document is considered as an (abstract) object with properties and relationships to other documents: If objects are considered regarding certain **DOCUMENT ATTRIBUTES** the interest is focused on special document features (e.g. author, class descriptor). This type of meta-information can be used for characterizing and distinguishing documents. An overview of document grouping and document/group relationships refers to **STRUCTURAL INFORMATION**. Many tasks consider information from documents and meta-information in combination.

Examples:

- **INFORMATION FROM DOCUMENTS:** Finding specific information about a topic.
- **DOCUMENT ATTRIBUTES:** Removing old documents from a collection.
- **STRUCTURAL INFORMATION:** Finding groups of documents with common features.

METHOD OF INTERACTION distinguishes tasks regarding their primary type of search. In an **EXPLORATIVE** method of interaction discovering interesting relationships and opening up the collection are in the fore. There is no clearly specified way towards the goal. Relationships among documents (implicit or explicit) and structural information can be used for orientation. In a **GOAL-DIRECTED** interaction the goal (documents or structures) can be described quite precisely. The focus of work is the identification of matching documents or structures.

Examples:

- **EXPLORATIVE:** Getting an overview of existing material.
- **GOAL-DIRECTED:** Identifying documents which match certain criteria.

GRANULARITY characterizes the level of detail of a task. The analyst might be interested in global and general relationships or coarse contents information (**OVERVIEW**) or in specific relationships and fine-grained information (**DETAILS**).

Examples:

- **OVERVIEW:** Defining a coarse grouping of documents.
- **DETAILS:** Understanding differences of highly similar documents.

CATEGORIES: Allows to describe in which way relationships between documents and classes are important. If categories are **RELEVANT** one option is to **USE EXISTING** categories in order to **CHECK CLASSES**: The maintenance and possible refinement of classes characterizes such a task. Another possibility is to **USE CLASSES** in the task: In this case a given class structure is exploited in some sense (not only for retrieval but also for comparing a given class structure against alternative structures and the like). If no categories are given it may be important to **CATEGORIZE** the collection.

Examples:

- **CHECK CLASSES:** Controlling the adequacy of class schemes.
- **USE CLASSES:** Find sparsely covered thematic fields.
- **CATEGORIZE:** Build a catalogue for use in a documentation system.

FOCUSED RELATIONSHIP deals with relationships that are important for solving the task. **DOCUMENT-SPECIFICATION** is an externally defined relationship: A specification is some kind of interest profile defined by the analyst, e.g. a query, a filter or a class definition. For the task it is important to which degree documents match the specification. In general, matching documents are in the focus of interest, but also non-matching items may be compared against the specification. The other relationships are **INHERENT**, i.e. they exist between items or groups of items within the collection. **DOCUMENT-DOCUMENT** relationships can be of diverse nature but they are usually based on some similarity or dissimilarity property. In a **DOCUMENT-TOPIC** relation the abstraction from documents to topics is relevant, whereas **TOPIC-TOPIC** describes relationships among topics which can be inherently defined by the group structure of documents.

Examples:

- **DOCUMENT-SPECIFICATION:** Checking consistency of a classification.
- **DOCUMENT-DOCUMENT:** Identifying clusters of similar documents.

- DOCUMENT–TOPIC: Identifying the topics of a collection.
- TOPIC–TOPIC: Analyzing the topical structure of a collection.

MODE OF COMMUNICATION characterizes the direction of communication which is dominant for the task considered. **RECOGNIZE** describes a cognitive reaction or association stimulated by presented items. The analyst reflects aspects of documents or the collection. Actively controlling the selection of document features to be analyzed is not in the fore. Rather, the analyst strives for a mental model of some collection aspects. On the other hand, **SPECIFYING** is related to actively controlling the presentation and selection of material to be analyzed (select, locate, define, focus, rank, and the like).

Examples:

- **RECOGNIZE**: Memorize or learn about the structure of the collection.
- **SPECIFY**: Define a categorization scheme for a document collection.

2.6.2 Application of the Model

The task model developed in this chapter provides a taxonomy for abstractly characterizing important features of complex analysis tasks concerned with document collections in knowledge management. It is meant as an aiding tool which is flexible enough to describe, classify and compare real-world analysis tasks. The following comments sketch the usage of the model and serve as a brief ‘best-practice guide’ for task characterization.

In the task model each task can be characterized by the relevant values of the eight criteria. For each criterion any combination of values is permitted, enabling to express that in a certain task some values may be valid at the same time. Dependencies among criteria or values are not modeled. In practice, however, some combinations will turn out to be more relevant than others. It is useful but not mandatory to use every criterion for describing a task. Note that when applying the model it is important to select those value combination which sufficiently describe all *dominant* facets of a task. Because the model has been constructed in order to characterize complex analysis task it is important to abstract from negligible details: It is not reasonable to describe every minor aspect but only characteristics which are essential to the task. Note that it is usually wise to decompose complex analysis tasks into coherent sub-tasks – if possible – in order to gain a better characterization.

Given a characterization of certain analysis tasks the model allows to group them regarding interesting criteria (e.g. regarding their dominant mode of communication or the resource which is primarily considered) and to define more complex types of tasks. The comparison of single tasks can make use of the hierarchical model structure: Since some of the criteria values are grouped it is possible to compare tasks on different levels of granularity. However, the degree of validity of criteria values depends on the concrete task considered and the degree of abstraction. Keeping this in mind the model allows to discuss differences and common grounds of tasks on a solid conceptual basis.

2.6.3 Describing and Comparing Tasks in the Model: An Example

As an example, consider two complex analysis tasks which will also be examined in the case studies (cf. chapter 8). The first is concerned with checking the single sources criterion in product documentation and aims at avoiding redundancy in a corpus of technical manual

documents (cf. table 2-6 and chapter 8.2), the second deals with exploring the term usage of a collection of research papers in order to support cooperative terminology work (cf. chapter 8.3). Though the second task is very complex and could be decomposed into sub-tasks it is considered as a whole in this example. Table 2-8 sketches the characterization of these two tasks in the proposed model.

Starting with the product documentation scenario, the goal is to figure out documents that provide the same or highly similar information in order to define single information sources and thus to *assure the quality* of a collection of documentation material: Redundancy has to be avoided since it might be a source of inconsistency, and documents may have to be edited accordingly. The *focus of interest* can be regarded as *fixed* on patterns of highly similar documents. *Information from documents* is examined and the analyst may have to *explore* the collection in order to find single-source candidates, taking advantage from the similarity *structure* of the corpus. Having identified such candidates the documents have to be considered in *detail*. The primary *relationship* which is interesting in this task is that *between* highly similar *documents*. The identification of single-source candidates is mainly done by *recognizing* patterns.

In the terminology work task users are interested in *learning* about the usage of terms in a collection of research papers in order to define a domain taxonomy. In a cooperative setting analysts extract commonly used terms from a text corpus and try to understand how concepts are used by different authors. The *focus of interest* is highly *adaptive*: Having found interesting terms, the meaning of a certain notion is discussed in a working group. This discussion, in

Table 2-8: Using the task model for classifying analysis task from case studies (see also chapter 8)

supporting terminology work by exploring the term usage in a collection of research papers			
checking the single source criterion in product documentation			
goal of interaction	making use of documents	learn	✓
		condense documents	
		select documents	
	maintaining documents	control quality	
		assure quality	✓
dynamics of focus of interest	fixed		✓
	adaptive		✓
resource considered	information from documents		✓
	meta-information	document attributes	
		structural information	✓
method of interaction	explorative		✓
	goal-directed		✓
granularity	overview		✓
	details		✓
categories	not relevant		✓
	relevant	use existing	
		check classes	
		use classes	✓
focused relationship	external: document – specification		
	inherent	document – document	✓
		document – topic	
		topic – topic	✓
mode of communication	recognize (reflect, physically passive)		✓
	specify (physically active)		✓

turn, stimulates other theories of term usage. Terms are *information from documents*. In order to understand the common usage of terms, the similarity *structure* of the collection is considered. Understanding term usage is an alternating process of *exploring* the notion of related documents and cross-checking in which other context identified terms play a role. The latter method of interaction is *goal-directed*. Both, *recognizing* and *specifying* are relevant modes of communication. Though some document details may be important the collection is mainly considered as a whole (*overview*). In order to quickly find out which author uses which terms a document-by-author classification can be regarded (*use classes*). Since this meta-information is not relevant for actually extracting terms, the value ‘document attributes’ is not chosen for the criterion ‘resource considered’. *Topical relationships* play an important role since common and differing term usage in different research areas is considered, and *inter-document relationships* are interesting for identifying texts that connect research areas regarding term usage.

Comparing these tasks reveals that for both structural information as well as information from documents is important. Both tasks rely (at least partly) on explorative data access and a means of recognizing patterns. However, the collections are considered on different levels of granularity: In the first case detailed document-to-document relationships are examined, in the second an overview of topical relationships is more important.

2.6.4 Reflecting the Task Model

The taxonomy developed in this chapter allows to describe real-world corpus analysis tasks – which play an important role especially in knowledge management – by their dominant features and typical facets. A classification of this model in comparison to other models is given in table 2-9. Considering the requirements defined in section 2.1 again, the definition of criteria and their values according to section 2.6.1 allows the *accurate use of the terminology*. The model is *flexible* since criteria values can be freely combined and considered on different levels of abstraction which allows to describe and classify tasks with different complexity. The taxonomy is *domain-specific* and optimized for characterizing analysis tasks which can potentially be supported by visual text-access interfaces: Being developed from task models for visual information retrieval and based on an intensive survey of important real-world tasks qualifies the model as being *extensive* with respect to the considered domain. Of course the model’s validation cannot be complete due to the complex nature of analysis tasks. It is not possible to rule out that some important and practically relevant tasks cannot be characterized sufficiently by the model. However, due to the method of model construction and the solid basis of practically relevant task samples, the model appears to be *plausible* and suitable as a tool for describing real-world tasks on a structured and abstract level. New criteria and values can be easily incorporated without effecting the structure of the remaining parts of the model, making it *extendable*.

Table 2-9: Classification of the resulting task model

model	modeled units	# units	domain	type of model
Shneiderman	system functions	7	domain-independent	set of units
Wehrend & Lewis	user actions	11	domain-independent	set of units
Belkin et al.	task features	4	domain-independent	feature space
specialized task model	task features	24	corpus analysis in KM	hierarchy

3 Methods for Text Representation and Computing Document Similarity

In typical text corpus analysis tasks the degree of relatedness of single documents or document groups is an important aspect. This chapter presents a brief overview of some document representation and comparison methods which have been proposed in the fields of information retrieval, information extraction and case-based reasoning. The aim is to examine the spectrum of different techniques for representing documents and for measuring document similarities or, more general, relatedness of texts for various application fields. Besides, section 3.1 introduces some technical terms which are assumed as known in the remainder of this thesis. In this work text analysis (or semantic document analysis) refers to the complete process of (a) generating formal document representations (called indexing) or identifying relevant information pieces in free text (such as authors or paper title in literature references), and (b) the computation of a degree of relatedness of documents (i.e. semantic similarity or dissimilarity of texts regarding a certain notion of relatedness).

The implicit hypothesis of most automatic methods for text comparison in specialized document collections is that determining text relatedness can be achieved by approaches (whether purely statistical or knowledge-based, whether using shallow or deep natural language understanding) which all depend directly or indirectly on the vocabulary used in the considered documents. Is there any linguistic affirmation of this assumption? Rieger [Rieg90] reports on findings which show that even very large corpora of *pragmatically homogeneous texts* contain only a very limited number of different lexical items, regardless of the personal active vocabulary of the authors. It is thus reasonable to expect that those terms which constitute specific information from a certain subject area are distributed within the considered documents, forming lexical-semantic regularities which can be derived empirically and statistically.

3.1 Vector-Based Methods from Information Retrieval

3.1.1 Indexing Terms and the Vector Space Model

The most prominent approach for representing documents is the ‘bag-of-words’ method where documents are described simply by a set of indexing terms (i.e. important words or phrases from the documents or descriptors from a controlled vocabulary). Indexing can be done either manually by experts who assign meaningful terms to texts usually from a controlled vocabulary, or automatically by computer-linguistic procedures and heuristics. In the latter case the text is broken into words, insignificant words (given by a so-called stop word list) are removed, and the remaining words are reduced to their stem. Formally, documents are seen as real-valued feature vectors over the indexing vocabulary of the collection. The weights of the terms (i.e. the corresponding values of the document feature vectors) characterize their importance for the document. Many different weighting schemes for indexing terms have been proposed in literature. The simplest case is a binary encoding where the value of a

component of the feature vector is 1 if and only if the corresponding term occurs in the document (this scheme is mainly used by the classic Boolean model of retrieval). More elaborated weighting schemes include the term frequency weights (tf), the $tf \times idf$ scheme where the term frequency is multiplied by the inverse document frequency in order to lower the influence of terms that occur in many documents, or weighting schemes based on information theory (*signal-to-noise ratio*, [Korf97]). In the vector space model of information retrieval [Salt71] both, documents and queries are seen as term vectors. Thus, not only queries can be matched against documents but also documents can be compared against each other. The retrieval function which compares queries and documents (or more general: the matching function that compares different term vectors) is based on vector algebra. There are many realizations of retrieval functions, e.g. the cosine measure of similarity (i.e. the dot-product of normalized term vectors) or the Euclidean distance as a measure of dissimilarity.

3.1.2 Latent Semantic Indexing and Concept Vectors

The simple ‘bag-of-words’ approach can lead to problems for document retrieval and comparison. On the one hand, many unrelated documents may share some indexing terms, on the other hand, documents related to the same concept may be indexed by different terms. As a result much noise is introduced during the computation of document similarities. A reaction to these term indexing problems is latent semantic indexing (LSI, [DDH90]). Latent semantic indexing is an algebraic model of information retrieval that is based on a singular value decomposition of the document-term matrix. As a result of the decomposition documents are represented in a concept-based space where the concepts are at a higher level than the indexing terms. The process of computing the semantic concept space which uncovers the latent semantic structure of the document-term matrix shall be briefly sketched: First, a document-term correlation matrix is constructed which stores the frequency tf_{ij} of a term i in document j . This matrix contains all information about term distribution within the considered document collection. The matrix is then decomposed into k (usually 100-300) orthogonal dimensions by singular value decomposition (a method related to eigenvector decomposition and factor analysis, see also chapter 4.1.2). In the resulting k dimensional vector space each ‘concept’ vector reflects the correlations in term usage across documents. Documents can be compared against each other using the dot-product of concept vectors (cf. cosine measure of similarity).

3.2 Text Analysis Based on Explicitly Defined Relationships

3.2.1 Citation Processing

The similarity of documents can also be measured by links that are explicitly given for a collection of texts. Virtually all scientific and many technical documents contain a clear indication of related material by the bibliography or lists of references that a document cites [Korf97]. In citation processing bibliographic reference links between documents are used as a measure of document-document similarity. For example, in co-citation analysis the similarity of documents is measured by the number of publications that cite both of the documents. Similarly, bibliographic coupling determines document relatedness by direct and indirect ‘citation links’. These heuristics or combinations of them may thus serve as application-specific measures of document similarity.

3.2.2 Hypertext Linkage

Similar to citations, hypertext links provide a direct kind of document connection. These links, usually created manually by the author of a document, point to material which is related to the source document in a certain way. Automatic methods for generating hypertext links have also been proposed [Allan95, Green99]. Analyzing the link structure of manually or automatically generated hypertexts can reveal some information about the similarity of the contained documents (cf. sections 5.2.2 and 5.3.1). In this context, though, it is necessary to distinguish semantic links and content similarity on the one hand, and structural links on the other, since both provide different information. A generalized similarity analysis between hypertext documents may incorporate both kinds of links and also include browsing patterns [Chen97].

Moreover, methods for automatically building hypertext links themselves rely on the computation of the semantic similarity of documents. Constructing useful links, however, requires a more accurate document (or paragraph) comparison than simple overall matching methods. This is due to the much finer granularity of linking paragraphs and documents: When a link is created from a sequence of words in one document to a different paragraph or text, the likelihood that the destination of the link is relevant has to be very high. Consequently, text representation and comparison approaches beyond standard indexing term vector similarity have been proposed. For example, Green [Green99] introduces a method based on lexical chaining (i.e. an approach for discovering sequences of related terms) where each document is represented by two vectors which provide different conceptual and relational information about the words that appear in the texts. The relatedness between a pair of documents or paragraphs is then measured by a combination of the similarities of the single vectors.

3.2.3 Information Retrieval and XML

A recent trend for representing and exchanging data on the Internet is to use the eXtensible Markup Language (XML, see www.w3.org/XML). This subset of SGML makes it easier to define new markup languages which enrich unstructured documents by semantic tags and structural information. Since it is widely expected that documents in XML format play an important role in future, information retrieval has started to address the question of how to exploit the structure of XML documents for information search: XML documents contain additional tags between which several relations can be defined. This information can then be used to improve query-driven retrieval (cf. [LCDL00, FuGr00]) or the computation of document similarity. In particular, context and relationship information can be used to process queries more precisely or to enhance the comparison of documents regarding their semantic similarity. Classic models of information retrieval, e.g. the vector-based methods (cf. section 3.1), ignore the structure of documents, so that the rich nested relationships between document parts get lost [ScNa00]. Since research has just started to address the topic of XML and information retrieval, many new results and applications for both, query-driven retrieval and document similarity matching are likely to be proposed in near future.

3.3 Text Understanding and Domain Knowledge

3.3.1 Structured Text Representations

Lewis [Lewis92] reviews several text representation methods from the viewpoint of document classification, among them methods for structured representations which go beyond a feature value encoding of documents. These methods explicitly include relationships among terms as part of the text representation itself. Examples of the discussed approaches include ‘relational indexing’ where roles between terms are modeled in order to allow queries which require that words with particular roles are present in the documents to be retrieved. Lewis reports that the particular method of relational indexing has never been implemented but was used for manual indexing. Similar structured representations, however, can also be constructed from texts using natural language processing. These representations then contain information about terms and their roles, e.g. as actors or objects, and types of actions described in the texts. Query or document matching is then defined as a partial matching of substructures of queries or documents. An application example for such representations is legal text retrieval. Another example for structured text encoding is based on conceptual graphs which allow for the use of well-founded inference methods for processing queries or matching documents. For a more detailed presentation see [Lewis92].

3.3.2 Information Extraction

Information extraction (IE, also known as message understanding) aims at extracting specific information – such as pre-specified types of events, entities or relationships – from unrestricted textual documents [RiLe94]. Thus it differs from information retrieval since it does not retrieve sets of potentially relevant documents based on key-word searching but fills in pre-structured templates with the desired information. In other words: The user receives exactly those pieces of information he has specified, not complete documents which may contain the desired information. IE systems are used for processing specialized document collections and for supporting restricted tasks, e.g. summarizing patient record in health care [LSA+94], scientific and technical literature monitoring, or monitoring newswire transcripts (a more detailed overview can be found in [GaRo97]). IE is computationally expensive and usually applied to not more than a few hundred documents. In order to apply it to larger and more heterogeneous collections it has to be coupled with traditional IR approaches that filter documents which might fall into the domain of specialized IE systems [GaRo97]. In order to overcome the limitation of IE systems to a priori fixed sets of templates, [HaRo00] propose a more general knowledge extraction system for technical documents. Here, conceptually and inferentially richer forms of relational information are extracted, and the set of templates is automatically enhanced by incremental concept learning. In any case, the extracted templates and or text knowledge bases can be used for different purposes, e.g. text mining, trend analysis, text summarization, document comparison, or simply for accessing specific information.

3.3.3 Textual Case-Based Reasoning

Case-based reasoning (CBR) is an important paradigm in automated reasoning and machine learning [LoPi97]. The idea behind CBR is to solve a new problem by considering similar, previously solved problems and by adapting their known solutions for the problem at hand in order to guide decision-making. Each CBR process involves the retrieval of relevant cases

from a case memory. These cases have to be represented by appropriate features for the considered application domain, and adequate similarity measures have to be defined. Instead of applying CBR to highly structured case descriptions, the relatively new field of textual CBR concentrates on applications where cases are available as texts, e.g. question-answer pairs in FAQ (frequently asked questions) lists [BHK95] or technical documentation [Lenz98a]. Textual CBR requires a very rich indexing of textual documents in order to allow reasoning. Obviously, this field has strong relations to information retrieval and information extraction. However, in contrast to classic methods of information retrieval, representing documents and assessing their similarity in textual CBR is based on sets of features established during knowledge acquisition and on similarity measures that incorporate domain theory [Lenz98b]. Researchers in the field of textual CBR strive for combined approaches that bridge information retrieval and CBR in order to support important tasks like realizing ‘corporate memories’ where a great deal of knowledge is contained in natural language documents [BFW98].

3.3.4 A Knowledge-Based Approach for Comparing Medical Abstracts

In [BST98] a knowledge-based method for indexing medical document abstracts and for assessing their similarity is presented. In order to gain an expressive indexing scheme for this restricted application field the approach makes use of a terminological knowledge representation language. One important observation in the application context is that disciplines like medicine are characterized by a certain “vague” notion, which means that medical facts can only be expressed insufficiently using a crisp logical representation. Therefore, an extension of classic terminological formalisms is used which allows vague concept interpretations, allows to define relationships with uncertain structure between concepts (so-called ‘roles’), and introduces vague attributes for concepts and roles [TBK+96].

The fundamental conceptual knowledge of the domain (i.e. a taxonomy of domain-specific concepts) and possibly vague attribute values for concepts and roles are defined in a knowledge base. In the application of medical abstract comparison the assertional component of the terminological system (the so-called ABox) is used for indexing the texts: Each document is represented as an independent ABox, i.e. a set of assertions of terms from the texts to concepts or roles, possibly connected with vague attributes. Thus, important concepts of a document along with their possibly vague relationships are formally described. One key problem is to extract instances of concepts and roles from a textual source, in this case medical abstracts that are written down in medical terminology. As a solution for this problem the adaptation of a medical terminus parser [TrTü98] that copes with specific characteristics of the medical terminology (medical findings in the example used in the cited paper) is proposed.

Finally, the document matching function uses a knowledge-based measure of similarity: For each pair of ABoxes a degree of similarity can be computed by comparing the contained concepts and roles, additionally taking into consideration their vague attribute values. This comparison process is supported by the structured information about concept relatedness: Using the definition of concepts from the knowledge base the subsumption relations of the conceptual knowledge can be computed. By doing this, a “vague” concept network is determined with the edges between concepts being weighted by their degree of similarity [TBK+96]. Thus, certain concepts are recognized as being more similar than others. The advantage of this knowledge-based comparison is that the resulting degree of similarity for documents is well-founded and probably more comprehensible for the user than a pure heuristic measure.

4 Methods for Visualizing Similarity Data

This chapter sketches a selection of methods for visualizing the similarity of data objects in the application field of exploratory data analysis. These techniques are assumed as known in the technical parts of this thesis. Section 4.1 presents methods for generating a coordinate-based representation of coordinate-free objects as well as for reducing the number of dimensions of high-dimensional data sets based on proximity information of the considered objects. Given that these methods are used to build one-, two-, or three-dimensional representations, the similarity structure of the corresponding data objects can be directly visualized using simple scatter plots. Alternatively, the constructed or reduced multi-dimensional representation can be used as input for other techniques which realize a graphical presentation of the similarity structure of high-dimensional data sets (see [KeKr96] for a survey of techniques for visualizing multidimensional, multivariate data). Section 4.2 sketches a completely different technique which maps high-dimensional data objects based on non-linear similarity relationships to a low-dimensional grid structure. Methods for visualizing the resulting arrangement of objects are also sketched. Chosen methods for the own approach, their realization and (where necessary) adaptation are presented in more detail in chapter 6.

4.1 Multidimensional Scaling and Related Techniques

Multidimensional scaling (MDS) is a collection of methods from multivariate statistical analysis which are used to construct a metric space of objects based on the observed proximity between each pair of objects [Davi83, Tulsa97, Math97]. A proximity measure δ for pairs of objects quantifies the degree to which the two objects are alike [Davi83]. If the highest value of this measure corresponds to objects which are most alike, δ is a measure of similarity. Vice versa, if the lowest value represents the highest similarity, then δ is called measure of dissimilarity. In the following, δ refers to a dissimilarity measure. In MDS a coordinate-based representation is constructed from coordinate-free dissimilarities. For a collection of n objects an arrangement of representatives in a m -dimensional space is determined so that the distances in this space (measured by a certain metric) approximate the observed dissimilarities of objects.

More precisely, given a set O of n objects, a dissimilarity measure $\delta: O \times O \rightarrow \mathbb{R}_0^+$ for each pair $i, j \in O$, and a target dimensionality m . Then, MDS constructs a set of vectors $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^m$ where each object $i \in O$ is represented by an $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$ with the property that the error s as defined by

$$s^2 =_{\text{def}} \sum_{i,j} (d_{ij} - \delta_{ij})^2 \quad (4-1)$$

is minimized [Tulsa97]. In the formula d_{ij} denotes the distance between i and j measured by a certain metric in the resulting m -dimensional space, and δ_{ij} is the observed dissimilarity of objects i and j ; s is called ‘stress’. The components x_{i1}, \dots, x_{im} of the object vector \mathbf{x}_i are called ‘coordinate estimates’.

MDS is a method of exploratory data analysis: For $m = 1, 2, 3$ these objects can be visualized so that an analyst gains insight into the structure of a set of data. This can help for detecting meaningful underlying dimensions of the data, thus allowing to explain the observed similarity or dissimilarity of objects. Of course, the definition of such discriminating features is an intellectual process. The applications of MDS are manifold (see [Davi83] for a detailed overview). Originally, MDS has been applied in behavioral and social science in order to describe and distinguish objects by some basic features. In computer science a similar application domain is using MDS in tools for knowledge acquisition [CoDo86]. Other applications include dimension reduction (representing k -dimensional data in an m -dimensional space where $m \ll k$), or data reduction (e.g. grouping of objects in a 2D or 3D space). In the following the basic methods of MDS and closely related techniques are sketched. Section 4.1.1 is mainly based on [Davi83].

4.1.1 Metric and Non-Metric Multidimensional Scaling Methods

Metric MDS approaches assume that the given dissimilarity data satisfies the properties of a metric, i.e. a function $d: O \times O \rightarrow \mathbb{R}^+_0$, where (a) $d(i, i) = 0$, (b) $d(i, j) = d(j, i)$, and (c) $d(i, j) \leq d(i, k) + d(k, j)$ for all i, j , and $k \in O$. One of the first MDS algorithms was proposed by Torgerson [Torg52]. This approach is even more restrictive and assumes that the dissimilarities are equal to distances in a Euclidean space. Torgerson's algorithm constructs the target space by applying a principal component analysis to a scalar product matrix which is directly derived from the input data.

Non-metric MDS is applied to dissimilarity data which does not satisfy the properties of a metric. Non-metric approaches attempt to reproduce the general rank-ordering of dissimilarities between objects as good as possible [Tulsa97]. More precisely, these MDS methods assume that the given dissimilarity data is related to metric distances in space only by an unknown monotone function f , i.e.

$$\delta_{ij} = f(d_{ij}) = f \left[\left(\sum_k |x_{ik} - x_{jk}|^p \right)^{1/p} \right] \quad (4-2)$$

such that

$$d_{ij} < d_{i'j'} \Rightarrow f(d_{ij}) < f(d_{i'j'}) \text{ for all } i, i', j, \text{ and } j'. \quad (4-3)$$

The metric used in equation (4-2) is a so-called L_p -metric (also known as Minkowski distance metric). Some algorithms require that the distance in the target space is the Euclidean distance (where $p = 2$ in equation (4-2)). The approach of Kruskal (1964) allows to choose any L_p -metric in the target space. However, non-metric MDS algorithms are generally fastest when the Euclidean distance ($p = 2$) is used. Moreover, less computational problems are encountered (such as getting stuck in local minima). When Euclidean distances are used, rotation of the space may be an issue, if the resulting space shall be interpreted by a researcher. This means that the user might have to rotate the space in order to get an interpretable solution.

According to [Davi83], each non-metric MDS algorithm consists of four phases (cf. figure 4-1). In the first phase a starting configuration of coordinate estimates is calculated, e.g. by applying a metric algorithm. Once such a starting configuration is determined, many algorithms employ some standardization of the current distances in the target space and the coordinate estimates constructed so far (for simplifying the solving of equations). In the non-metric phase the dissimilarity data and the preliminary results are used to compute so-called

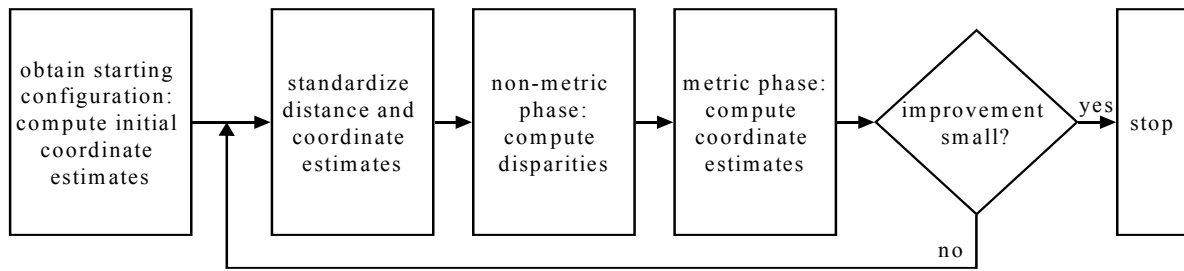


Figure 4-1: Flow chart of non-metric MDS algorithms [Davi83]

disparities, i.e. rank images δ'_{ij} of the original dissimilarity data δ_{ij} , the values of which match the distance estimates as good as possible, but which respect the following constraint:

$$\delta_{ij} < \delta_{i'j'} \Rightarrow \delta'_{ij} \leq \delta'_{i'j'} \text{ for all } i, i', j, \text{ and } j'. \quad (4-4)$$

Only after these disparities have been calculated, new coordinate estimates are computed in the metric phase, minimizing the misfit of reproducing the rank order of the δ_{ij} . If there is still space for improvement, the resulting arrangement of objects is the input for the next iteration step. The iteration stops when the improvement in fit falls below a suitably small threshold.

4.1.2 Multidimensional Scaling versus Factor Analysis

Factor analysis is a technique which is mainly used for reducing the number of variables (data reduction) or for detecting structures and relationships between variables [Tulsa97]. In the latter sense the research question addressed sounds similar to that of MDS, though both methods are fundamentally different. In general, MDS is by far less restrictive. While factor analysis techniques require a correlation matrix of certain attributes for which the analysis is performed, MDS can be applied to any kind of proximity data (not only correlation coefficients) as long as the rank-ordering of values in the given matrix is related to some distance measure in a metric space in the sense expressed in equation (4-3) [Tulsa97]. An application example for factor analysis in information retrieval can be found in chapter 3.1.2.

4.1.3 Force-Directed Placement

A different solution to the object arrangement problem has been applied mainly in the field of graph drawing [Ead84, KaKa89, FrRe91]: Force-directed placement is a dynamic and iterative method which applies force laws from physics for arranging objects in a low-dimensional space (whether for weighted graphs or for object relationships described by similarity or dissimilarity). Corresponding approaches interpret object similarities or dissimilarities as physical forces and try to find a force-balanced state which reflects the structure of the input data. More precisely, the main idea of these heuristics is to simulate physical models where particles move according to force and energy effects and come to a halt in positions where the energy sum is minimal [Sand96]. The objects to be placed (graph nodes or other abstract objects) are regarded as particles, the weights or similarities are considered as forces or energy. Though many different formulas for forces are applied, the principal is the same for all approaches: In general, these methods start with an arbitrary initial configuration, simulate the movements of the particles and lower the energy in each step until the particles come to rest [Sand96]. An early example of a force and energy controlled placement approach is spring

embedding. This technique models object relatedness by mechanical springs of different strength and rest length and applies spring formulas for finding a force-balanced solution.

4.1.4 Geometric Scaling

MDS is a complex analytic procedure which requires a computation time of $O(n^3)$ for placing n objects. Moreover, when changes in the set of objects are made or objects are added to the set, MDS requires a re-calculation of the complete arrangement. In [FaLi95] a scaling method (called FastMap) based on a geometric mapping is introduced which provides the following basic features: It realizes a fast scaling of n objects into k dimensions in time $O(kn)$, preserves the dissimilarity information fairly well, and allows to quickly add new objects to the space in time $\Theta(k)$ without having to re-calculate the entire space of objects. However, it requires that the given dissimilarity data satisfies the properties of a metric. Obviously, FastMap does not require the computation or definition of all $O(n^2)$ distances between the objects under consideration.

The basic idea of the algorithm is to interpret the objects to be mapped as points in a multi-dimensional space with unknown dimensionality and to successively map them onto k mutually orthogonal axes of a Cartesian coordinate system by only using the given distance information. Compared to MDS, the quality of the results is inferior for a fixed number of objects to be mapped and a fixed number of dimensions of the target space. However, the proposed algorithm achieves dramatic time savings over MDS, with respect to both, the number of objects and the number of dimensions [FaLi95]. Given a fixed computation time, on the other hand, FastMap is able to produce qualitatively superior results (price/performance relationship). Summing up, the FastMap algorithm is a good choice for a fast arrangement of objects and for supporting retrieval tasks in databases since it accelerates the search time for queries, though it needs more dimensions in the target space to yield a solution quality comparable to that of MDS. Another advantage of this method is that is by far easier to implement than the methods discussed above.

4.1.5 Scatter Plots for Visualizing Scaled Data Sets

Given n objects in a k -dimensional coordinate-based representation, for $k = 1, 2$, and 3 and a reasonable size of n , the natural way of graphically presenting the data is to use a scatter plot. Scatter plots are used in exploratory data analysis to display coordinate-based objects on typical Cartesian (perpendicular) axes. They are a diagnostic tool for determining association of

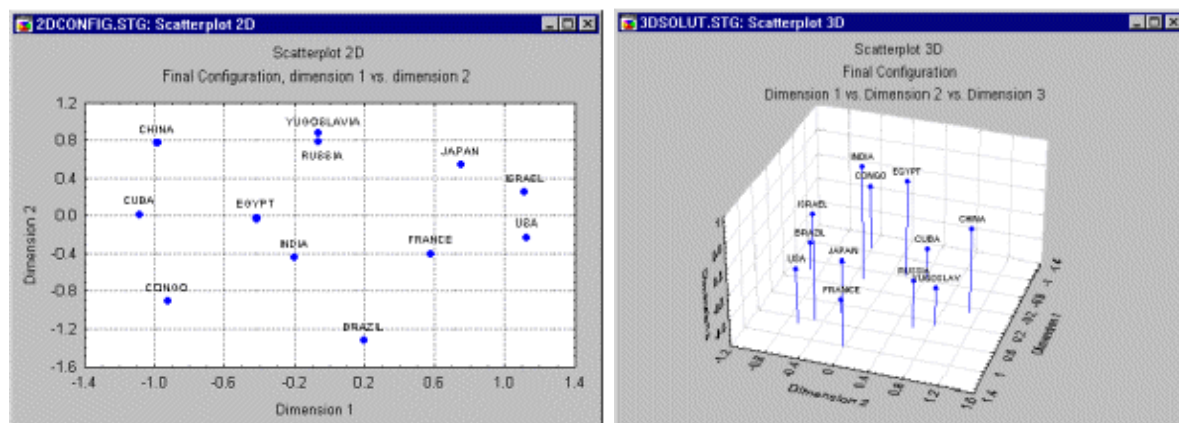


Figure 4-2: Scatter plots (2D and 3D) of data sets scaled by MDS techniques [Tulsa97]

data points (objects) in the given data set [HeFi01]. Relationships of some kind appear in the plot as any kind of non-random structure. For classic MDS applications in social sciences or market analysis, applying scatter plots is a first step for interpreting the dimensions. Sometimes, the plot has to be rotated in a certain direction (either manually or automatically) in order to gain an interpretable solution. For applications in information visualization (cf. section 5.4) it is often not necessary or possible to extract features for describing the data though the data's similarity can be meaningfully assessed. Then, the dimensions do not obtain any explicit semantics but are used to express the relative relatedness of objects based on a certain assessment on their similarity. This is done, for example, in order to group objects and thereby to perform data reduction or just to find patterns of related objects. Figure 4-2 shows two scatter plots of a very small data set.

4.2 Self-Organizing Feature Maps and their Visualization

A self-organizing feature map (SOM) [Koho82, Koho95] is a simple neural network model, proposed by Kohonen, with the ability to realize a mapping of nonlinear statistical relationships between high-dimensional input vectors to a (usually) two-dimensional grid-structure of network units. The mapping, which is realized by an unsupervised learning algorithm, preserves most of the topological information of the input data, in particular the cluster structure. In other words, a self-organizing feature map orders vectors from an input space according to their similarity in two dimensions. The SOM algorithm combines unsupervised competitive learning with dimensionality reduction by smoothing clusters of the input space with respect to a given output grid.

4.2.1 The Model of Self-Organizing Feature Maps in a Nutshell

Self-organizing feature maps belong to the class of single-layered feed-forward networks [Zell94]. They consist of one layer of active units which are organized in a d -dimensional grid structure. In most cases a two-dimensional regular grid is used, but other layouts (one-dimensional structures and rarely grids with three or more dimensions) have been used, too. The units in the grid are not connected with each other. A layer of input units comprises as many units as there are dimensions in the input space. Each grid unit i is linked with all units of the input layer by means of weighted edges, formally realized by a weight vector \mathbf{w}_i . Initially, random numbers are assigned to the components of this weight vector. The output function calculated by each unit measures the distance or similarity between the input pattern \mathbf{x} and the unit's weight vector \mathbf{w}_i .

During the unsupervised competitive learning process, a single unit i^* is determined for each input vector at each time t so that its weight vector is most similar to the given input pattern \mathbf{x} . Such a winning unit i^* is called the *cluster center* (or codebook vector) of the input vector \mathbf{x} . The weight vector of the cluster center and the weight vectors of units i in a certain surrounding of i^* , determined by a time-dependent neighborhood function v , now are shifted towards the input vector according to

$$\mathbf{w}_i(t+1) =_{\text{def}} \mathbf{w}_i(t) + \alpha(t) \cdot v(i, i^*, t) \cdot (\mathbf{x} - \mathbf{w}_i(t)). \quad (4-5)$$

Thus, the amount of shifting depends on a time-dependent learning rate α , the distance between weight vector and input vector, and the unit's position in the area surrounding the cluster center i^* according to v . Both, learning rate and the size of the area defined by the neighborhood function decrease in time.

4.2.2 Properties of the Model and Comparison with Scaling Methods

As an effect of the learning process, weight vectors are ordered according to their similarity in the grid of units. Furthermore, the distribution of weight vectors reflects the density of the input space. After the learning process of the SOM any two clusters that are close to each other in the grid have cluster centers which are close to each other in the input space. It is important to note that the converse does not hold: Cluster centers that are closely neighbored in the input space do not necessarily correspond to clusters that are close to each other in the grid. To phrase it another way: A SOM tries to embed the (output) grid in the input space in a way that the cluster centers are close to the corresponding patterns in the input space without having to stretch or twist the grid too much [Koho95]. In this sense the SOM realizes a topology preserving mapping which tries to capture the neighborhood relations of objects in the input space on the output grid of units. Depending on the number of network units used, units either represent means of corresponding clusters of input data (less units than input patterns, leading to data reduction), or they represent single input patterns and generalized data (at least as much units as input patterns).

There is a certain relation to the MDS methods and related approaches from section 4.1: MDS produces a non-linear mapping from a high dimensional representation of objects (possibly from a certain space with unknown dimensionality where only the distances or dissimilarities of objects are known) to a metric space with low dimensionality. In contrast to the *topology* preserving mapping of the SOM, the kind of mapping realized by MDS is *distance* preserving (cf. [KLHK98]). In favor of preserving the overall neighborhood relations, a SOM does *not* try to preserve distances. As a consequence, a SOM is not suitable for faithfully mapping geometric objects from a high-dimensional space into an output space with lower dimension (e.g. for pattern recognition). Rather it helps for clustering, visualization and abstraction [Koho95].

4.2.3 Visualization of Self-Organizing Maps

Vesanto [Vesa99] identifies three categories of visualization methods for self-organizing feature maps based on the goal of visualization, namely (a) getting an idea of the cluster structure of the input space, (b) analyzing correlations between input vector components, and (c) examination of new data samples for tasks like classification. This section concentrates on the first category which can be further divided into techniques for displaying the shape of the data clouds in a space with reduced dimensionality (*projection methods*) and methods for showing clusters on the SOM grid (*distance matrix methods*). Projection methods visualize the *weight vectors of the SOM* usually in a two- or three-dimensional space. These weight vectors form data clouds in the input space. A way of displaying the shape of data clouds is to use the SOM for data reduction and to project the weight vectors of units into a two- or three-dimensional space by using techniques of MDS. Such projections can give a rough idea of clusters in the data [Vesa99].

When *clusters on the SOM grid* shall be displayed (e.g. for a presentation of neighborhood relationships rather than displaying absolute distances of weight vectors, cf. section 4.2.2), distance matrix techniques are typically used. These approaches exploit the SOM property that weight vectors are ordered according to their similarity and that the distribution of weight vectors reflects the density of the input space. The *U-matrix* approach (*unified distance matrices*, [UISi90, Ult93]) is probably the most prominent distance matrix technique. This method determines the distance of each unit's weight vector to the weight vectors of its neighboring units and visualizes the resulting value matrix by shades of gray: the higher the distance, the

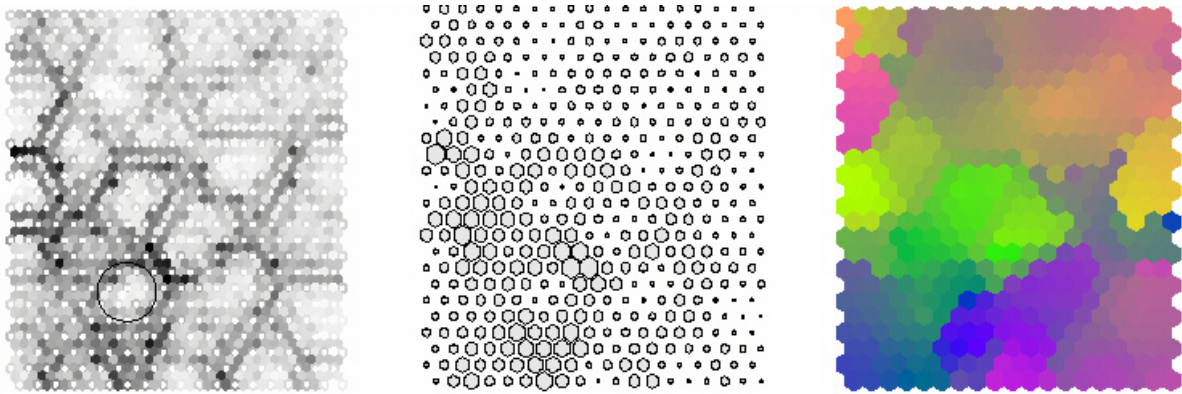


Figure 4-3: Different distance matrix visualizations of a SOM for the same data set: (a) U-matrix, (b) distance matrix, and (c) similarity coloring (source: [Vesa99])

darker the gray shade. Figure 4-3(a) shows an example of this visualization method: White dots represent SOM units, and gray-shaded hexagons between them indicate the similarity between each pair of neighbored weight vectors. Bright areas in the resulting display correspond to clusters in the input space, separated by dark areas. A similar visualization approach (cf. [Vesa99]) is to calculate for each map unit's weight vector the mean, median, maximum or minimum of its distances to all of its neighboring units' weight vectors, and to visualize the respective units by circles of sizes that are proportional to the averaged distance value. In this sense an averaged version of the *U*-matrix is gained (see figure 4-3(b) for an example). The third picture in figure 4-3 shows a similarity coloring of the map units where units are coded with similar colors if the corresponding weight vectors are close to each other in the input space.

Summing up, both, the *U*-matrix and the distance matrix approach calculate and visualize the distance between each pair of weight vectors that correspond to neighbored units of the output grid. Though the resulting visualizations help to effectively recognize structures in the input space, there are some drawbacks coming along with only considering neighbored units [SBJ99]: First, the quality of the visualization strongly depends on the distribution of weight vectors in the input space. A good quality of the distance matrix approaches sketched above requires that the weight vectors mainly reproduce the input patterns, i.e. the network must be intensively trained, leading to a tendency for over-generalization. Second, there is no immediate relation to the input patterns themselves, thus leading to a more 'abstract' view on the input space.

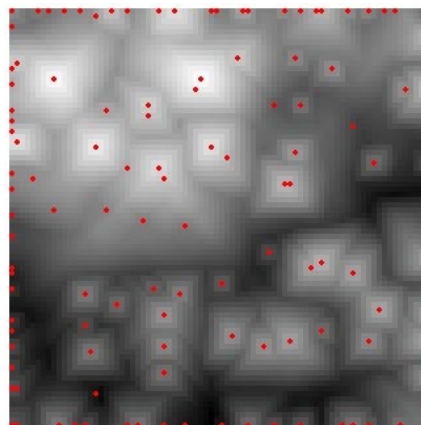


Figure 4-4: SOM visualization based on the *P*-matrix method (the data set differs from that used in figure 4-3)

An alternative approach, proposed by Sklorz [Sklo96, SBJ99], is the so-called *P*-matrix visualization method. Instead of calculating the similarity only between neighboring units, this technique determines the value of the greatest similarity between the weight vector of a considered unit and all sample patterns from the input space. Similar to the averaged *U*-matrices, these values – which correspond to single units of the grid – can be visualized by shades of gray (for a more detailed presentation cf. section 6.2.3). Figure 4-4 shows a SOM visualization of a *P*-matrix. Each gray value corresponds to the *P*-matrix value of a single unit; cluster centers of input patterns are marked by the dots in the display. The remaining units represent generalized data patterns. Clusters in the input space are separated by dark borders, and the strength of the separation corresponds to the relative distance of the clusters. As a consequence of the alternative similarity assessment, structures in the input space can be detected in the visualization independent from the achieved degree of generalization. Moreover, each visualized grid unit has a meaning which is directly related to the patterns in the input space, thus allowing a fine-granular analysis of the space with respect to the input data. These properties are particularly interesting in the application domain examined in this thesis.

5 Structuring and Visualization Techniques for Document Collections

“Give the searcher something to look at!”

(Lauren B. Doyle, 1961)

Information retrieval systems aim at enabling a content-based and query-driven search in document collections. However, there is a wide spectrum of possible search types due to the fact that the focus of interest may be narrow or cover wide areas of the document base. While in the first case the user has a pretty good understanding of what he or she is looking for, in the latter case the need for structure and semantic ‘search paths’ for a document collection arises. Closely connected to the demand for structuring a corpus of documents is the desire to visualize the often complex relationships between documents or classes of documents. As vision is a human’s most powerful sense when it comes up to discovering patterns, anomalies or trends – or simply finding one’s way to a possibly complex search goal – graphically displaying a collection’s structure is a straightforward consequence.

This chapter reviews methods for structuring and visualizing document collections which support explorative approaches to the collection’s contents. Although the focus is on methods which visualize a certain similarity of documents within a corpus some further classic and popular work will be presented. In particular categorization and hypertext organization will be discussed which both provide a suitable basis for visualizing structure.

5.1 Semantic Structure for Guiding Information Access

In chapter 2.2 the notions of goal-directed and explorative document access have been introduced. These search types are characterized by the degree to which a respective search goal can be specified. In general, the necessity for guidance and structure clearly grows with the degree of vagueness of the information need (see figure 5-1). When relevant documents cannot be explicitly described on the basis of the search goal some kind of structure has to be provided to allow fruitful access to the document base: Exploration requires structure! Doyle, in his classic article ‘Semantic Road Maps for Literature Searchers’ [Doyle61], goes even further and points out the importance of ‘familiar conceptual grooves’ in document search in general:

“[...] it is probably psychologically sound to give a literature searcher some kind of structure, whether hierarchical or otherwise, rather than to give him an infinitely and therefore totally disorganized array of terms.”

Today, there are many different approaches in information retrieval which aim at providing a certain structure for browsing document collections and for presenting this structure intuitively to the user. The goals are manifold, reaching from easing the identification of relevant documents in goal-directed searches up to enabling a collection’s exploration and mining tex-

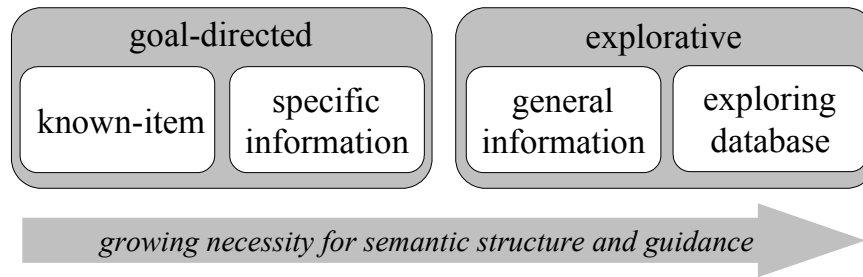


Figure 5-1: Search types and the need for structuring document collections

tual databases. The vivid research field of information visualization for document management and retrieval is still rapidly growing.

5.1.1 Focus of Interest

Figure 5-2 schematically presents the context of the approaches to be presented in this chapter. Mainly, all these methods support explorative search types. Among the classic methods for providing explicit structures are hypertext linking and categorization of documents (section 5.2) which both may serve as a basis for visualization. There are various aspects of individual documents or document collections which can be graphically presented: Link maps visualize the structure of hypertexts, thus pointing out certain relationships between different documents. Category maps graphically present the association structure of document classes. Some visual retrieval interfaces concentrate on selected attributes of a given set of documents, others display the topic or term distribution of individual documents. Both directions can be of help in the context of graphically presenting query results – an intersection of goal-directed search and exploration. Paragraph 5.3 gives a brief overview of approaches related to this group. This chapter's emphasis is put on the discussion of document maps which graphically present the overall structure of a corpus of documents (sections 5.4 and 5.5). For automatically structuring document collections cluster analysis is very important. Therefore, the following section briefly sketches the role of clustering in information retrieval.

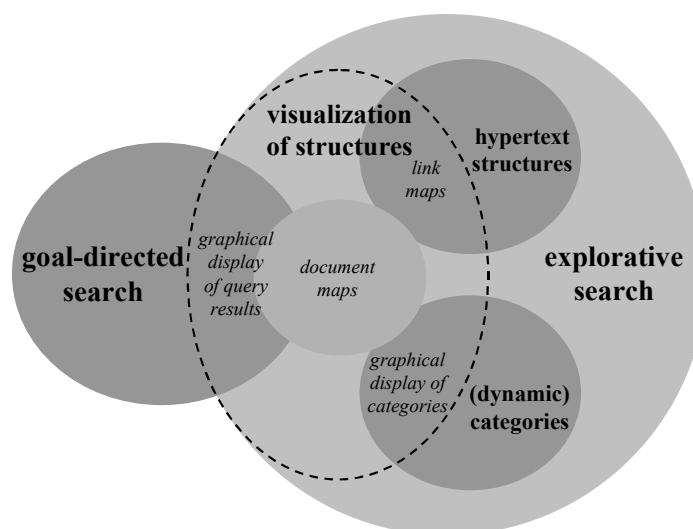


Figure 5-2: Context of methods to be presented in this chapter

5.1.2 Background: Clustering in Information Retrieval

In the context of information retrieval clustering has been studied extensively (a review can be found in [Will88]). Its role is twofold: Clustering *terms* was investigated e.g. for the classification of textual documents [CHO+94] or for automatically constructing thesauri [Chen94]. A related approach can be found in computer-linguistics where representational word meanings were examined by cluster analysis in a semantic word space: In such a space linguistically labeled elements represent meaning-points and their mutual distances represent meaning differences [Rieg83, Rieg85]. Clustering *documents* (which is the focus of this section) can be seen as an extension of the idea of ‘semantic closeness and distance’ of single terms. It was originally examined to improve the *efficiency* of query processing, i.e. to reduce the number of documents that need to be compared to a given query [Salt71]: In such a document clustering approach the query is matched against cluster representatives first (e.g. the centroids of the clusters); only for promising clusters the included documents are considered further.

C.J. van Rijsbergen put forward the idea to improve the *effectiveness* of retrieval by grouping similar documents and, thus, taking advantage of inherent semantic relationships between single documents. In his *cluster hypothesis* he states that closely associated documents tend to be relevant to the same requests [Rijs79]. In other words: The similarity of relevant documents among each other is higher than the similarity of other subsets of the collection. Following this assumption (which can be experimentally tested for a given collection) he proposed *cluster-based retrieval* which returns entire clusters in response to a query in order to improve recall.

Voorhees [Voor85] critically discussed the cluster hypothesis and concluded that it frequently does not hold and that even if it is true cluster-based retrieval often performs poorer than sequential search, i.e. retrieving *individual* documents with respect to the query. However, in many publications various cluster-oriented techniques were proposed as an alternative or extensional retrieval method and search strategy. They do not aim at retrieving entire clusters of documents but rather use clustering as a retrieval technique in its own right for different purposes: Some approaches directly group similar or somehow related documents in order to generate a (possibly temporary or virtual) physical structure for browsing (cf. sections 5.2, 5.5.1, and 5.5.2). Others visualize the similarity structure of a collection and thus graphically present clusters of documents (cf. sections 5.3, 5.4, and 5.5.3).

Clustering is always based on some measure of similarity between the objects to be grouped. Similarity, in general, means that a certain relationship between the attributes of different objects holds to a certain degree. Various reasonable relationships between documents can be defined: Besides the classic measures of information retrieval – which are based on term occurrence – certain link structures between documents may be useful. For example, citation links for scientific documents have been studied early (see e.g. [Small73]). Another approach is to measure the relatedness of Web documents on the basis of hyperlink structures (cf. sections 5.2.2 and 5.3.1).

Yet, to be confined to similarity alone is no dramatic restriction. As Korfhage states “distance alone matters” [Korf97]: In query-driven retrieval similarity (or distance) is the common measure of relevance. Documents are ordered relating to decreasing similarity (or increasing distance) to the query. Thus, the query is a ‘point of reference’ for a one-dimensional spatial arrangement. This situation can easily be generalized to multiple reference points (i.e. multiple queries or interest profiles, respectively). Consequently, documents themselves can be regarded as reference points for each other document. Mapping this high-dimensional arrangement to a low-dimensional space is the basis for visualizing the (similarity) structure of document collections.

5.2 Structuring Document Collections for Exploration

This section is concerned with methods for building explicit or detecting implicit structures of document collections in order to allow content-based retrieval and exploration. Some classic approaches of the field are sketched due to their relevance in the context of semantically structuring specialized document collections.

5.2.1 Categorization and Dynamic Clustering

The traditional approach in Information Science for structuring large document collections is the use of catalogues which are organized according to classes of documents. Therefore it is necessary to categorize documents with respect to their contents. In its general definition *document categorization* (also referred to as document classification) is the problem of assigning documents to a number of pre-defined and non-disjoint classes, usually representing subject headings. Besides describing the content of documents in a controlled and unambiguous way categories usually are grouped and thus expose some content structure of a collection. In the terminology of statistics categorization is related to discriminant analysis. In contrast, *document clustering* is concerned with the creation of meaningful ‘natural’ classes, prior to the assignment of documents to these classes. More precisely, the heterogeneous set of documents is divided into homogeneous groups according to the documents’ features. These groups are finally labeled with class descriptors. In this sense clustering precedes classification. However, the notion of document categorization is used ambiguously in literature, namely for both, classification and clustering (cf. [Lewis91]). From the structuring point of view classification impresses an ‘external’ structure on the collection whereas clustering figures out its inherent, ‘natural’ structure. Sections 5.5.1 and 5.5.2 present methods for building classes based on clustering. Due to the terminology used by the respective authors the notion ‘categorization’ is used although these approaches detect a collection’s inherent structure rather than organizing a set of documents according to given categories.

Traditionally, in libraries documents are categorized manually using documentation languages. The concept of *documentation languages* in general comprises all approaches which provide a controlled vocabulary for describing the contents of a document. Among them automated indexing methods which use thesauri or conceptual languages from Artificial Intelligence (e.g. [Schm94]) can be found. Classification systems are the traditional form of documentation languages. In Computer Science the *Computing Classification System* (CCS) – formerly known as *Computing Reviews Classification System* (CRCS) of the ACM journal *Computing Reviews* – is a very prominent example [CR97]. In this system subject headings from a hierarchy of topics are assigned to documents. For example, the subject heading H.3.3 refers to the topic ‘Information Search and Retrieval’. Furthermore, so-called general terms can be used to build facets of the topic, e.g. ‘algorithms’, ‘performance’ or ‘human factors’. Finally, free terms can be assigned to describe the document more detailed. In practice, most documents are indexed by multiple CCS descriptors.

In information retrieval automatic methods for the classification problem have been intensively studied. Just to point out the complexity of the problem two examples shall be given: [BHMP92] assigns documents to a small number of disjoint classes. Based on a sample set of documents – classified by a human expert – this approach generates a rule system for performing classification of new documents. To set up the rule base techniques from Natural Language Processing are used for indexing the documents and discriminant analysis is applied for calculating a probability of membership for each document and each class. In

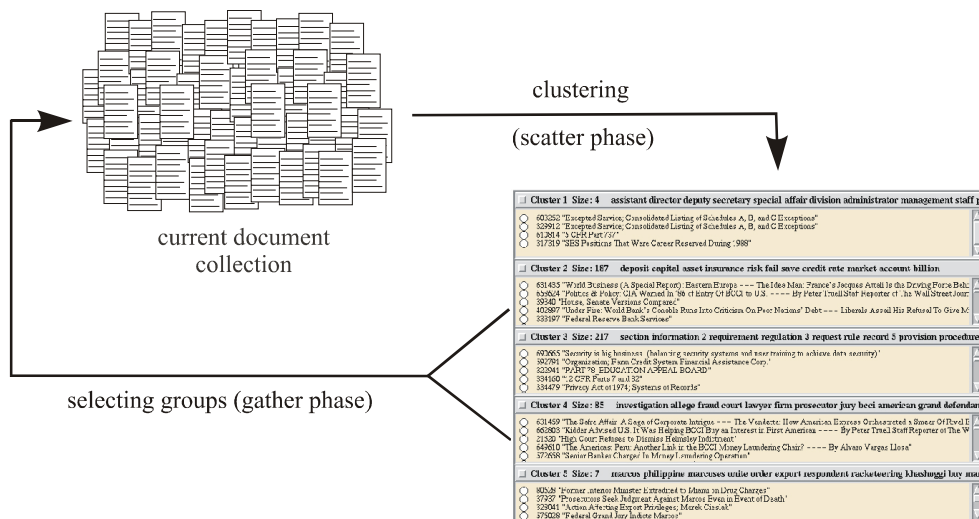


Figure 5-3: Interactive cluster browsing

[LeGa94] a probabilistic classifier is trained in an iterative process. This machine learning approach stepwise refines an initial classifier by asking a human teacher to label documents for which the classifier is least certain of class membership.

A prominent cluster-based retrieval technique is scatter/gather browsing [CKPT92]. This approach aims at providing a means for gradually narrowing the focus of a search and is based on iteratively clustering a collection. The main idea of scatter/gather is to dynamically present to the user descriptions of groups containing related documents. The clusters are constructed according to dominant key words (scatter phase). Among these groups the searcher can select interesting ones which are then combined (gather phase) and re-clustered. The browsing process stops when the user switches to a focused (goal-directed) search or picks out single documents. Figure 5-3 depicts the interactive process of scatter/gather browsing. The interface presents the detected clusters, described by a so-called *cluster digest* which contains the dominant key words of the cluster along with the titles of the group's documents. The user may select single documents or groups for further processing. Scatter/gather requires fast clustering techniques to allow its application to large corpora of documents [CKP93]. It was shown to yield an idea of a collection's structure and content [PSHD96]. Applied to result sets of query-driven document retrieval (cf. section 5.3.3) this method has proven to significantly improve the identification of relevant documents over ranked result lists alone [HePe96].

5.2.2 Hypertext Structures

The World Wide Web can be seen as the most popular means of organizing large collections of documents. Developed to be a 'pool of human knowledge' [BCL+94] it consists of a loosely and highly collaboratively organized collection of documents. Each individual person can add information and link it to other existing resources by creating a hyperlink. A *hyperlink* points from one document to another one which is related in some way to the source object. Furthermore, the density of links between certain nodes may reveal interesting information about the 'closeness' of linked objects. However, the relationships between documents created by hyperlinking are usually of an unknown type, i.e. the semantics of a link can be understood only by human interpretation of linked items. Nonetheless, hypertext structures are a special case of semantic structuring, albeit the structures in practice are merely blurred by different and inconsistent meanings. In addition, the *lost in hyperspace* phenomenon

arises: Users may not know where they are within the huge structure and where exactly they can get to from a certain point. There are several ways to enhance the structures of and the navigation in hypertext systems; some of them are presented in this section. The visualization of hypertext structures will be discussed in section 5.3.1.

The ‘natural way’ to search for information in hypertext systems is to browse through the structures, following hyperlinks which possibly lead to the information source desired by the user. The meaning of a hyperlink, however, is given only implicitly by the surrounding information in most current HTML-based systems. This circumstance can be improved by the Extensible Markup Language XML [Haro98] which allows to mark up information according to its meaning. Besides, there are several research efforts which aim at enriching hypertext structures with semantics. For example, Wang and Rada [WaRa98] propose a structured hypertext system for technical documents which contains structural and relational rules for defining hyperlinks between documents. The system enforces these constraints when hyperlinks are constructed by the user. Domain semantics are defined on the basis of semantic networks and reflect structural and semantic characteristics of the underlying documents. Allan [Allan96] proposes to automatically annotate links with their corresponding type from a pre-specified link taxonomy to give the user some idea of the relationship between the source and the destination of a link. The taxonomy includes link types like ‘summary’ (the destination object contains a condensed discussion of the topic) or ‘tangent’ (the destination objects contains further information of only tangential interest).

The latter method is based on an approach of automatically constructing hypertexts from a given input collection by using techniques from information retrieval [Allan95]. This research effort was motivated by the expectation that with the rapidly growing number of documents in the World Wide Web a purely manual construction of large hypertext structures becomes problematic. In the proposed method documents with a sufficiently high similarity are decomposed into pieces (e.g. sentences, paragraphs, sections or phrases). A link between corresponding pieces of different documents is established when the similarity of the regarding text segments exceeds a certain threshold.

Hypertext structures contain and link documents of many different topic areas. It is thus important to flexibly define individual semantic sub-structures of the Web resources. For manually defining semantics over hypertext structures so-called ‘structured maps’ [DMRA97] can be used. They allow to embed documents in a semantic network and support a customized use of the information. Browsing assistants are used to automatically ‘personalize’ hypertext structures with regard to the interest profile of the user [JoMI98]. An example for this is WEBWATCHER [JFM97], an automatic tour guide for the WWW which observes and learns from its user’s actions. Based on its observations the tool judges the respective relevance of documents within the collection. Links that are assumed to lead to resources of interest are highlighted by the software agent. The goal of the WebCluster project [MHM98] is to allow a user to generate an individual view to the Web by filtering documents which are considered as relevant with respect to a given reference collection.

Another way of improving browsing and retrieval in the often loosely correlated structures of hypertext systems is to perform cluster analysis on these structures, thereby increasing the degree of structuring. Hypertext structures implicitly induce a measure of similarity: Links between nodes, whether direct or via link paths, indicate a certain relationship among documents. Using the ‘density’ of links between single documents the approach of [Bota93] generates clusters of Web pages. These clusters can be used for simplifying graphical displays of hypertext structures (cf. section 5.3.1) or for browsing through meta-structures before switching to a detailed hypertext browsing session.

5.3 Visualizing the Structure of a Text Corpus: An Overview

Graphically displaying complex information directly appeals to the powerful human visual perception which enables the user to rapidly identify patterns, trends and anomalies in large amounts of data. Consequently, visualization turns out to be a valuable tool for managing the growing mass of data and information available. Gershon and Eick state that “[...] a key research problem is [...] to discover new visual metaphors for representing information and to understand what analysis task they support” [GeEi98]. For information retrieval and document management visualization provides a means of easily identifying outliers, boundaries and clusters in the document space, thus pointing out the structure of heterogeneous document collections. Furthermore, the context of interesting documents within a collection can be easily explored if a suitable metaphor for the structure’s graphical display is provided. This section gives a brief overview of the various directions of visualization for different tasks in the field of managing documents.

5.3.1 Visualizing Hypertext Structures

The World Wide Web hypertext structure allows easy navigation across distributed information sources. The information space defined by the link structure, however, is explicitly hidden to the user. Yet, this structure contains valuable information about the relatedness of different sites. For example, the hyperlink structure between different research sites may reveal interesting information about strategic partnerships or dominant research issues. Furthermore, in addition to the approaches discussed in section 5.2.2, the presentation of the topological structure of a hypertext can aid goal-directed navigation and help to overcome the *lost in hyperspace* phenomenon.

As creating links in a hypertext corpus can be seen as a special case of semantic structuring – resulting in a graph of text nodes – (cf. section 5.2.2), visualizing hypertext structures is closely related to the visualization of the inherent semantic structure of document collections. The problem of how to visualize complex hypertext linking structures is a vivid research issue. This section therefore briefly sketches some important work of the corresponding field of research. The approaches presented here are concerned with generating a more general view on larger hypermedia structures rather than constructing navigational assistance on a strictly local level (for graphical tools focused on the document level cf. [GWL+95]).

Dieberger [Dieb95] discusses some spatial navigation strategies for the World Wide Web, including maps of the geographical structure of Web servers or virtual landscapes. The latter are constructed by Web users in a shared environment and contain graphical objects that represent links to WWW pages. Objects related to similar topics are manually placed close to each other.

The system NARCISSUS [HDWB95] visualizes link structures between objects, e.g. Web pages or links in technical manuals, by applying a self-organizing method which is loosely based upon a physical system with forces acting between objects. Link structures exert attractive forces between objects which leads to a close clustering of strongly connected documents. In the display single objects are presented as space balls using virtual reality techniques. Another system which can be used for visualizing link structures is VXINSIGHT which is discussed in more detail in section 5.4.3. Its method for building up the document space for visualization is similar to that of NARCISSUS. In contrast, it is not restricted to hypertext collections. Figure 5-4 shows the graphical representation of the hyperlink structure of different research sites, created by VXINSIGHT.

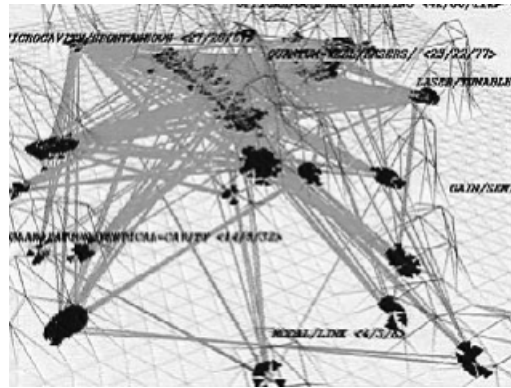


Figure 5-4: Hyperlink structure of different research centers [VxHP99]

The NAVIGATIONAL VIEW BUILDER [MuFo95] is a tool which allows to interactively create certain graph-oriented views (overview diagrams) of a hypertext structure. Due to the fact that a direct visualization of the link structure would be very complex and does not reveal any information about the actual contents of the nodes the tool incorporates different abstraction strategies. It is based on database-oriented hypermedia systems, i.e. hypertext nodes along with some meta-information (including manually inserted attributes regarding the page's topic) are stored in a structured database. The hypermedia structure can be clustered or filtered either regarding its link structure or its content, i.e. nodes containing links of a certain type or attributes which satisfy certain properties are clustered together or filtered for presentation, respectively.

5.3.2 Visual Information Retrieval Interfaces

The interest in visual information retrieval interfaces (VIRIs) has grown rapidly in the past decade. One reason for this is the availability of cheap and powerful graphical hardware. This broad and large field is characterized by a considerable diversity, both regarding the style of presentation as well as the aims and goals (for an overview and some example systems see [Korf97]).

A coarse distinction can be made by the granularity of the systems: Some systems focus on the individual document level. For example, TOPIC ISLANDS [MWBF98] visualizes the 'thematic flow' of a document and allows the analysis of its topical characteristics. A second group of interfaces pursues the goal of revealing some partial structure of the document space, among them approaches which figure out the structure of query result sets (cf. section 5.3.3) and systems which are concerned with relationships between documents according to some user defined dimensions. For example, VIBE (Visual Information Browsing Environment, [Dubin95]) is a multi-reference point system which provides a 2-dimensional display for presenting the relationship of documents regarding to user-selected reference points. These reference points represent terms which can be positioned on the edge of a displayed circle by the user. Based on the significance of documents with respect to the terms document icons are arranged on the display. The size of the document icon relates to the importance of the document with respect to the reference point.

Finally, there are interfaces which aim at conveying 'a picture of the whole': Document collections are visualized using some spatial metaphor. Closeness of items in these spaces corresponds to 'relatedness' of the associated documents. Interfaces of this category will be presented in the sections 5.4 and 5.5 of this chapter.

5.3.3 Displaying the Structure of Query Result Sets

The group of work described in this section aims at providing visual methods for a better identification of relevant documents in query results. Query result sets are usually presented as ranked lists of references to matching documents. The motivation behind these approaches is to reveal some sort of structure of result sets, thus making them accessible more intuitively and improving retrieval effectiveness. In particular, a basic assumption is that outliers, i.e. documents wrongly judged relevant, can be identified more easily by visualizing relationships of retrieved items. The visualization of a query result set's structure can be related to different levels of detail.

ENVISION [NFH+96] is a system which graphically presents a variety of document characteristics and attributes on a two-dimensional display. It allows, e.g., to differentiate retrieved documents regarding attributes like *author* and *publication date, year* and relevance value or indexing terms and relevance value. Thus, different views on the result set can be generated.

Other approaches allow to compare the structure of individual documents with respect to query terms: For example, TILEBARS [Hea95] graphically displays the relative length, query term frequency and query term distribution of retrieved documents. Each document from the result set is visualized using a rectangle the length of which corresponds to the document length. The rectangle consists of multiple gray scaled squares, one for each textual paragraph, and the darkness of each square indicates the query term frequency in the corresponding segment. The method proposed in [VeHe97] allows to inspect the distribution of query terms in retrieved documents by means of displaying a matrix of frequency bars for each query term and each document from the result set.

Finally, there are methods which focus on the overall similarity structure of query result sets: [WuWi98] propose a clustering of the retrieved documents and use a textual display to present the cluster structure. CAT-A-CONE [HeKa97] uses a graphical cone-tree representation of category labels. Categories which are associated with the retrieved documents are highlighted in the tree. By interacting with the result set and the category hierarchy the user can explore the semantic structure of the retrieved documents. In [ALS97] a spatial display of the relationship between retrieved documents is proposed. Documents are represented as balls in 2- or 3-space. Their distance reflects the similarity of all pairs of documents from the result set. In [SwAl96] the authors refer to this graphical representation, which is generated by using a multi-dimensional scaling approach, as 'document map' (cf. sections 5.4 and 5.5). The initial spatial arrangement of documents can be improved interactively by relevance feedback techniques (cf. [Salt71]). Using the relevance feedback information provided by the user the display is modified by bringing relevant documents closer together.

5.4 Document Maps and Landscapes

This section sketches approaches for visualizing the similarity structure of a document collection by means of a certain graphical metaphor, such as a star field display or a landscape of mountains and valleys. There is no common notion for this kind of graphical presentation. Some authors call their visualization method 'document landscapes'. Others use the notion of 'document maps', stressing the fact that the graphical representation of landscapes or star fields is reminiscent of geographical or astronomical cartography. For most visual metaphors used in the context of this chapter the notion of 'document maps' seems suitable. While this section gives an overview of various document map approaches based on numerical scaling

and clustering techniques, section 5.5 concentrates on approaches which use a self-organizing neural network mapping to generate the graphical display.

In general, the idea of document maps is to use a familiar graphical representation in order to convey information about the relationships of individual documents or groups of documents, respectively. By doing this, the highly complex multidimensional document space is reduced to a two- or three-dimensional spatial representation which thereby enables a user to grasp the information space, thus making it accessible for more intuitive analysis or exploration.

5.4.1 BEAD

BEAD [ChCh92] is a prototype system for the graphically-based exploration of bibliographic data, developed at Rank Xerox Cambridge EuroPARC. It presents similarity relationships between documents by the relative spatial position of corresponding ‘particles’ in a 3-dimensional space: Close particles represent similar documents, dissimilar documents correspond to particles placed in some distance. This mapping process results in 3-dimensional point clouds. Searching for keywords in the documents leads to a highlighting of particles related to matching documents in the display. The system was tested using a corpus of approximately 300 abstracts from articles related to human-computer interaction.

The method of calculating the spatial arrangement of the documents – an approach related to multidimensional scaling, cf. chapter 4.1 – is a combination of techniques from information retrieval, numerical optimization and computational physics: Documents are represented as particles in 3-space. Based on the characteristics of the documents a potential field between the particles is set up which reflects the difference between the actual geometric distance $d(a,b)$ of particles a and b and the desired document distance $\delta(\alpha,\beta)$ for documents α and β , where a and b are the corresponding particles for α and β . The document distance measure $\delta(\alpha,\beta)$ is adopted from information retrieval. The ‘physical behavior’ of these particles is defined by a force which is linearly proportional to $\delta(\alpha,\beta) - d(a,b)$. This produces an N-body problem. By applying methods from numerical optimization to the particle system, namely steepest descent and simulated annealing, the unbalanced force on each particle is minimized. In this way a position for each particle is found which best reflects the document similarity in 3-space.

In internal and informal experiments conducted at EuroPARC it was found that the 3-dimensional cloud representation of documents does not easily yield an overview of the entire collection. Therefore, in [Chal93] the graphical display is improved by using a landscape metaphor. This graphical presentation can be compared to a view of a landscape from an observation tower which is, according to the authors, more 2.1-dimensional than 3-dimensional and thus results in a more familiar spatial presentation than real 3D. Individual documents, now, are represented as colored markers placed within the setting of the landscape regarding to their similarity. Matching documents in a keyword-search are highlighted by changing the color of the markers.

5.4.2 SPIRE

At Pacific Northwest National Laboratory the SPIRE system has been developed [WTP+95]. SPIRE (Spatial Paradigm for Information Retrieval and Exploration) is a system suite, originally developed for the U.S. intelligence community, which comprises several tools for visualizing relationships of textual documents on different levels [PNNL99]. Two technologies within SPIRE, both based on term vector representations of the documents, are aimed at visu-

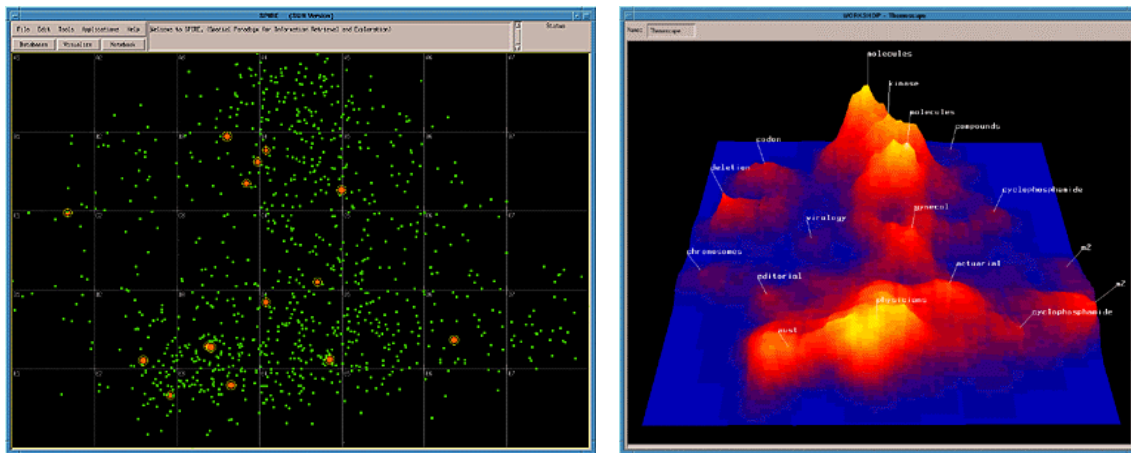


Figure 5-5: GALAXIES: scatter plot of documents (left), THEMESCAPE: distribution of topics (right) [www.cartia.com]

alizing the document space (figure 5-5): GALAXIES (cf. [HeMi98]) yields a simple 2-dimensional scatter plot that positions documents based on the similarity of their content. The metaphor used for visualization is that of ‘stars in a night sky’, i.e. documents are represented as points on a dark background, and closely related documents are clustered together. THEMESCAPE [HHHW98, HWMT98] graphically displays the distribution of dominant themes in a collection by generating a landscape of mountains and valleys. Instead of directly representing single documents the relative strength of topics within the collection is visualized by peaks of different height: the higher the peak, the stronger the topic. Similar topics are located near each other. Both, the projection methods of GALAXIES and THEMESCAPE are based on multidimensional scaling.

Since 1996, researchers from the SPIRE team produce document landscape interfaces for commercial use in a PNNL spin-off (originally named THEMEDIA, the company is now called CARTIA), based on SPIRE technology.

5.4.3 VxINSIGHT

VXINSIGHT [DHJ+98], developed at Sandia National Laboratories, Albuquerque, is a graphical tool for the contextual examination of query results and for database exploration. The system's core component is a 3-dimensional graphical display which visualizes similarities among objects (e.g. papers pre-selected by an SQL-query against a database of scientific documents) and their distribution density by means of a mountain terrain metaphor, very similar to that of SPIRE's THEMESCAPE. In contrast to SPIRE, the system is a general purpose tool for different types of objects from structured databases. It visualizes certain relationships of the objects and allows to query the database using an SQL interface. Matching data elements are highlighted in the graphical display of the object collection's structure. VXINSIGHT provides a multi-resolution viewing capability, i.e. the user can zoom into an area of interest to see more structure.

The development of VXINSIGHT was motivated by the need to analyze research literature for supporting decisions about the placement of new scientific projects and for planning strategic partnerships. The analyst can choose among different object similarity functions, such as common keywords in documents, citation links in scientific papers or direct links in Web documents. By drawing lines between documents the citation or direct links can be graphically presented (cf. section 5.3.1).

The construction of the landscape display is done on the fly. Thus, fast visualization techniques rather than powerful but time-consuming approaches like self-organizing maps (cf.

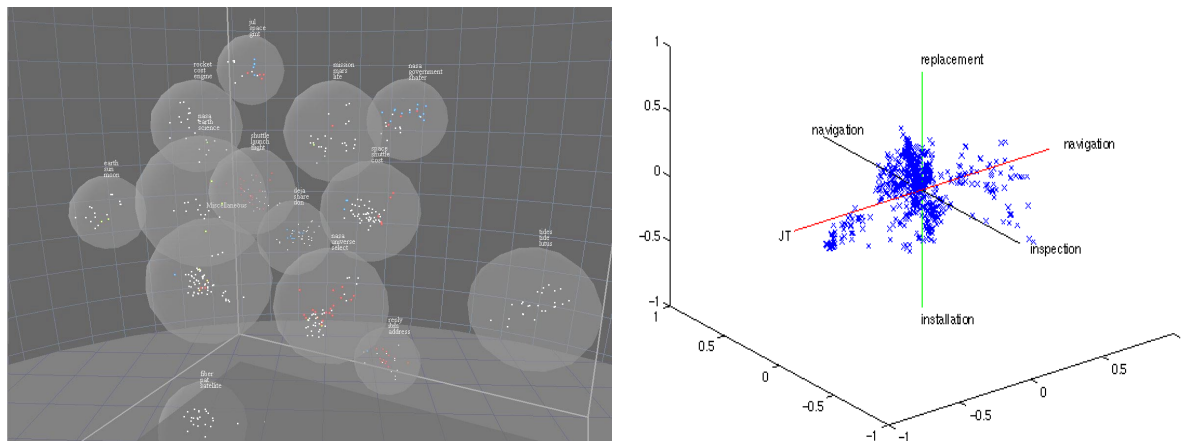


Figure 5-6: (a) Starlight Visualization Screen, (b) Subspace representation in TRUST [BCGH+99]

section 5.5) are used. For calculating the graphical display the system contains two ordination algorithms. The first approach, related to multidimensional scaling, involves eigenvectors of a Laplacian matrix and has the property to find the global minimizer of the penalty function which measures the stress between intended and real distance of objects. Its drawback is the tendency to produce too tightly clustered displays. The second algorithm is a so-called force directed placement (cf. BEAD, section 5.4.1) which produces more appealing visualizations for the cost of tending to get stuck in local minima. By first applying the eigenvector approach and refining its result by force directed placement the individual drawbacks of each single method are lowered.

5.4.4 STARLIGHT / TRUST

STARLIGHT [RMDT96] is an information visualization system under development at the Pacific Northwest National Laboratory (PNNL). The project is competing with PNNL's SPIRE (cf. section 5.4.2). Originally designed for analyzing intelligence data STARLIGHT aims to visualize the content of multimedia databases. It yields a 3D scatter plot representing the proximity of the data sets. Yet, each non-text object is described by a unique text file as a common basis for display processing. STARLIGHT's visualization approach, similar to that used in BEAD (cf. section 5.4.1), applies a 3D metric multidimensional scaling technique. Figure 5-6 (a) shows a scatter plot of data from the 'sci.astronomy' newsgroup. A clustering algorithm yields the displayed spheres around the document clusters.

Since 1996 PNNL and The Boeing Company collaboratively work on the development of STARLIGHT [HGKP98]. While PNNL provides the graphical capability for visualizing text data sets Boeing's Applied Research & Technology group contributes a text processing tool-set for performing indexing and querying operations on textual databases. In particular, Boeing's proprietary text mining engine TRUST (Text Representation Using Subspace Transformation) is used for document analysis [BCGH+99]. This system was designed with an emphasis on user interoperability and personalized data exploration. In TRUST documents are represented as term vectors. In order to reduce the dimension of the resulting vectors, a method called 'orthogonal decomposition' is used. Here, the idea is to find so-called content subspaces where a set of terms forms a concept. More precisely, orthogonal decomposition creates new features derived by linear combinations of terms. Each concept can then be represented by an axis. The user can select interesting axes and documents are displayed regarding their position in this subspace (figure 5-6 (b)). The decomposition technique is a variation of

Latent Semantic Indexing [DDH90] that has been modified in order to allow these user-defined perspectives.

5.4.5 KNOWLEDGE GARDEN

KNOWLEDGE GARDEN [CDMR98] is a collaborative information visualization tool, developed at British Telecom Laboratories. Here, the goal is to provide a virtual environment where users can meet each other and share relevant information. Each user collects documents from the World Wide Web, adds meta-information, and files them as bookmarks in a shared information store. Besides automatically informing other users about new documents which match their interest profile the Web pages are locally indexed. By applying a hierarchical clustering algorithm similar documents are grouped [DWRM96]. The metaphor used for the graphical display of document groups is that of flowers growing in a garden. Each flower represents a cluster of related documents. Stalks of a flower represent single documents, a colored icon at the end of each stalk indicates the status of the document (e.g. ‘document updated’ or ‘link dead’). Thus, the structure and status of a collaboratively collected document corpus is visualized by the landscape metaphor of a flower garden. The document garden is constructed using the Virtual Reality Modeling Language (VRML).

5.5 Self-Organizing Document Maps

Self-organizing feature maps (SOM, cf. section 4.2) realize a mapping of nonlinear statistical relationships between high-dimensional input vectors to a two-dimensional grid structure. The mapping, which is realized by an unsupervised learning algorithm, preserves most of the topological information of the input data, in particular the cluster structure. An important application area of this model is the analysis and visualization of high-dimensional data, e.g. in engineering applications and data mining [SVAH99, Vesa99, Sklo96]. In recent years, self-organizing maps were adapted for purposes of structuring and visualizing textual document collections.

5.5.1 Category Maps

Lin was probably the first to use a self-organizing feature map for document retrieval. He proposes the application of Kohonen’s model for building a category map as a retrieval interface for an online bibliographic system [LSM91, Lin92] and uses the metaphor of a ‘graphical table of contents’ [Lin96]. The idea of his approach is to provide a visual display that shows the contents and the associative structure of document collections in terms of neighborhood of categories. First applied to small collections and title-based indexing he later used his approach with larger collections and full-text indexing [Lin97].

Figure 5-7 depicts a category map (adopted from [LSM91]) of 140 documents indexed by the descriptor ‘Artificial Intelligence’ from the LISA database. The map is divided into so-called concept areas, formed by dominant key-words. The resulting classification is a partition of the set of documents, i.e. each document is assigned to exactly one area (class). Each area’s size corresponds to the frequency of the labeling terms in the document collection. Association of classes is expressed by the neighborhood of concept areas: Due to the topology preservation property of self-organizing maps frequently co-occurring terms form concept areas which are located near each other.

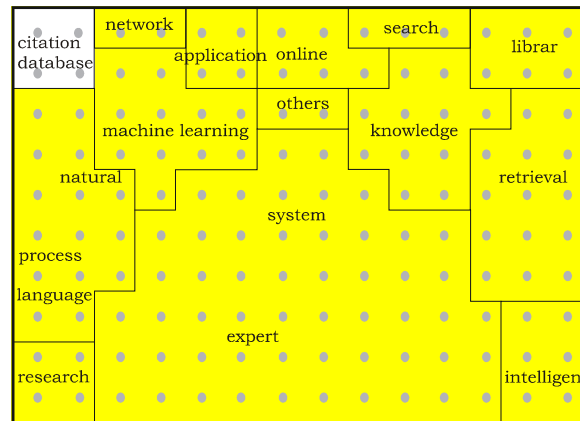


Figure 5-7: Category map after Lin et al. [LSM91]

The user of category maps can get an impression of what kinds of topics can be found in a collection (which justifies the ‘table of content’-metaphor). Furthermore, related topics can be found close to each other, and dominant topics can be identified by the size of the corresponding concept area (the map shows that in the late eighties ‘expert systems’ must have been very prominent in AI literature).

The basic method of generating a category map consists of the following steps:

1. *Encode the documents as term vectors.* Term vectors consist of real-valued components representing the weight of a corresponding index term. These terms can be generated automatically from the input documents by removing common words, reducing the remaining terms to their stem and eliminating most and least frequent words. Weights can be computed, according to Salton’s heuristics for weighting in the Vector Space Model [Salt71], by counting the frequency of each term (tf). Additionally, the inverse document frequency (idf) can be applied which computes the inverse of the number of documents in which a term occurs. For training the self-organizing map it is crucial to produce low-dimensional input data. Thus, the generation of indexes is based only on titles, keywords provided by the authors, and abstracts.
2. *Train the network using the generated term vectors.* After this process the network will map similar term vectors (according to a certain similarity measure) to neighbored units of its grid structure, thus reflecting the structure of the input data. The topological information of the input space is encoded in weight vectors associated with each unit.
3. *Derive concept areas from the trained network and present them graphically.* Concept areas are coherent parts of the grid which represent dominant keywords. They are determined by computing the most similar unit vector – i.e. a vector with weight 1 at exactly one position, thus representing a single term – for each of the network’s weight vectors. The unit is labeled with the dominant term. Neighbored units with identical labels form conceptual (term) regions. Finally, concept areas of frequently co-occurring terms can be merged.

A very similar approach was made by Chen et al. [CSO96] who address the problem of providing a ‘concept-based categorization and search capability’ for the World Wide Web and apply a SOM for automatically categorizing WWW-homepages. In contrast to Lin, who builds ‘tables of contents’ for relatively small document collections, Chen aims at classifying

Table 5-1: Some important technical details of Lin's and Chen's category map approaches

		Lin et al. [LSM91]	Lin [Lin96, Lin97]	Chen et al. [CSO96]
document space	Number of documents	140	143 – 660	several hundred – 10,000
	Parts of documents used for indexing	titles	titles, keywords, abstracts	full
	Number of indexing terms (input vector dimension)	25	85 – 1472	≈ 1000
	weighting scheme	binary	binary, tf, tf \times idf	binary
	Dimensionality reduction	none	none	use only most frequent terms
SOM	Size of SOM (number of units)	10×14	$10 \times 14 / 14 \times 14$	10×20
	Training cycles	2500	2500	≥ 5 times per document
	map layers	single	single	multiple (usually 5–6)

“millions of homepages according to their content”. To achieve this he modifies the map generation method as follows:

1. *Document encoding*: To ensure an appropriate (i.e. relatively low) dimensionality for the neural network's input vectors use only the n (e.g. $n = 1000$) most frequent terms as indexing vocabulary. To scale up the method for higher numbers of indexing terms in [RoCh98] a modification of the SOM-algorithm for binary input vectors is introduced which takes advantage of the typical sparseness of the documents' index-vectors.
2. *Recursive application for large concept areas*: For regions containing more than k (e.g. $k = 100$) documents take the corresponding documents as input for the calculation of a new map. This results in a hierarchy of category maps, thus forming multiple map layers. Chen states that 5–6 layers should be enough to “partition Internet resources into meaningful and manageable sizes”. Table 5-1 compares some important technical details of Lin's and Chen's approach.

In a comparative study Lin examined the automatically generated map display and human-generated displays and found that both “provide reasonable structure to show underlying document relationships” [Lin95]. Chen found that the contribution of category maps lies in assisting broad information searches, i.e. in situations where subjects have to skip around certain categories [CHSS98] (cf. section 5.6).

Roussinov [RoRa98, Rous99a, Rous99b] has extended Chen's category map interface in order to provide an ‘adaptive visualization’, i.e. to enable the user to modify some aspects of the category visualization, such as removing terms, rebuilding maps using more specific terms or reducing the number of regions in the map. These features allows the user to influence category generation and add a kind of ‘personal view’ on category maps.

more	.	format	.	diskcopy	.	time	.	mem
.	.	chkdsk	.	diskcomp	.	date	.	.
.	mirror	.	assign	.	unformat	.	.	type
recover	undelete	.	.	.
.	.	.	mkdir	.	del	.	find	fc
ren	.	rmdir	comp
.	.	.	.	copy	.	append	path	.
attrib	.	cis	.	restore	.	xcopy	replace	tree
.
edit	.	edlin	.	backup	.	dir	chdir	join

Figure 5-8: A map of MS-DOS commands [Merk195c]

5.5.2 Classification Aided by Self-Organizing Document Maps

Merk1 introduced a SOM-approach for structuring software libraries according to the semantic similarities of the stored software objects' descriptions [MTK94, Merk195a]. The aim of this approach is to provide easy access to libraries for system developers who want to retrieve software components for reuse. Figure 5-8 presents a map of MS-DOS commands where similar commands are located near each other. In this simple grid representation each position corresponds to exactly one grid unit of the feature map. The labeled entries mark units which represent documents.

Such a map is intended to serve as an interface to the underlying software library [Merk195c]. However, the identification of clusters is left completely to the user who would have to mark regions of related documents. Thus, these maps aid the classification task by a simple organization of related documents to nearby grid units. They do not provide any additional information on how to group the items. In [WSM96] this map approach is applied to 41 text segments of legal documents, pre-selected by the query term 'neutrality', in order to aid the separation of different meanings of this technical term for knowledge acquisition.

The maps are generated by training a self-organizing map using binary term vectors derived from the software descriptions. Each document is finally mapped to the grid unit which is most similar to the documents' term vector. Table 5-2 gives some technical details. In order to

Table 5-2: Technical details of software component maps (cf. [MTK94])

document space	number of documents	36
	parts of documents used for indexing	command descriptions
	number of indexing terms (input vector dimension)	39
	weighting scheme	binary
	dimensionality reduction	none (later: multi-layer perceptron)
SOM	size of SOM (number of units)	12×12
	training cycles	30,000
	map layers	single

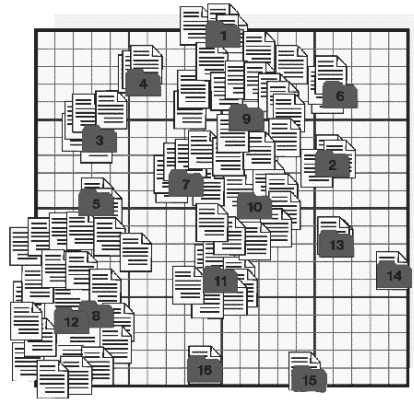


Figure 5-9: Classification map (reprinted from ESA Bulletin No. 87)

accelerate the training of the SOM when using long term vectors Merkl proposes an input vector compression (dimension reduction) generated by a three-layer-perceptron prior to building the map [Merkl95b]. Though he has shown that training the SOM with compressed data yields the same structuring in shorter time, he did not take into calculation the usually high amount of time necessary for training the perceptron. An improvement for the map display was made in [Merkl97a] where a so-called *adaptive coordinates* technique for visualization is introduced: Starting with the initial regular grid the coordinates of each unit are moved with regard to the relative distance of their weight vectors, thus reflecting some of the similarity information encoded in the trained network.

A variation of this document map approach is given in [Merkl97b] where a hierarchical feature map (cf. [Miik90]) is applied for classification. In such a hierarchical map setting each grid unit of a non-bottom layer corresponds to a further SOM. After training one layer each SOM on the next layer is trained only with the feature vectors mapped to their corresponding unit from the higher layer. Thus, multiple documents have to be assigned to each single grid unit, which are then considered as a group of related documents to be refined in the respective map of the next layer. Quite clearly, the number of layers and grid units for each map directly influences the number of classes or sub-classes, respectively, to be detected on every level. Consequently, a suitable and careful design of the hierarchical map's architecture requires some *a priori* insight into the collection's structure.

Another approach which uses hierarchical SOMs for aiding the classification of documents is described in [TrWa96]. This research activity – carried out at ESRIN, the European Space Agency's (ESA) Centre in Frascati, responsible for non-operational data processing and information systems – examines different neural network approaches for information retrieval tasks like query expansion and document classification, e.g. unsupervised Hebbian learning and multi-layer perceptrons for principal component analysis. For performing the classification task Kohonen's network, again, has proven to be successful. Figure 5-9 shows a map presenting the relative distance of top-level clusters in the ESA Microgravity Database – a collection of 975 documents which describe all European life and physical science experiments carried out on ESA and NASA missions. The map is intended to help the maintainers of the database to perform document classification.

For encoding the documents the ESA project applies a Hebbian network: The documents are first coded as term vectors using the occurrence count of word stems, produced by using a full-form dictionary (i.e. a dictionary containing each grammatical derivation of a non-stop-word and its corresponding stem). Using a training set of these document description an unsupervised Hebbian network is trained as a means of dimensionality reduction. In the example

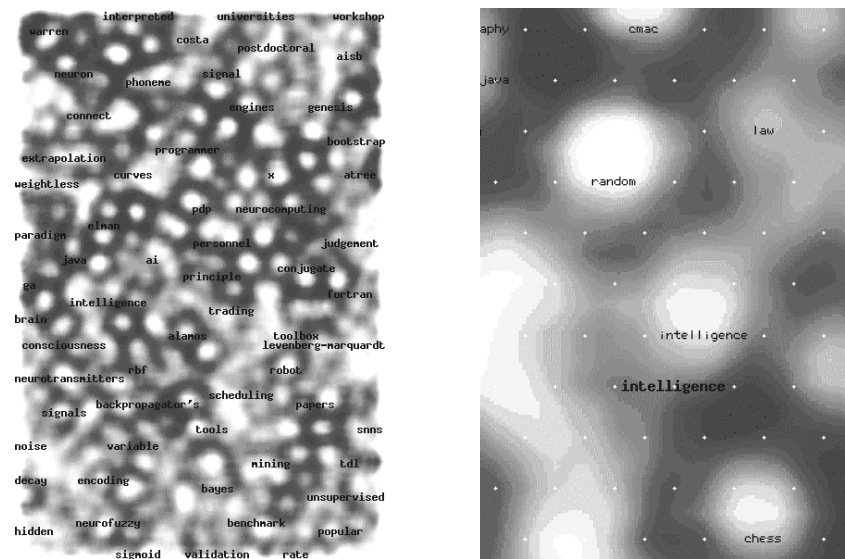


Figure 5-10: (a) Map of newsgroup articles from comp.ai.neural-nets (b) Sub-map of area 'intelligence' [WebSOM]

given by the authors the original term vectors consist of 2962 dimensions, the resulting 'semantic pattern vector' contains 100 elements.

For generating the classification map, a hierarchical SOM is used to produce clusters of related documents. An iterative clustering is performed only for those clusters which exceed a certain size predefined by the user. Labels, derived from a group's dominant key words, are assigned to each cluster. For example, in figure 5-9 cluster 1 contains 56 documents and is labeled 'texas, setup, rocket, series, spacelab'.

5.5.3 WEBSOM

Whereas category maps (cf. section 5.5.1) are topic navigation maps which focus on labeled classes of documents the WEBSOM-project ([HKLK96, LHKK96, HKLK97, KHLK98, Koho98], lead by T. Kohonen) aims at developing a 'landscape of document collections' where the overall similarity of texts within a collection is visualized. Documents dealing with similar topics tend to be located near each other in a two-dimensional grid representation of the document space. Each grid unit contains a set of very closely related documents. The primary goal of the work is to provide an interface for browsing through collections of newsgroup articles.

Figure 5-10 shows a WEBSOM-map of 12,088 newsgroup articles and a sub-map magnifying the area of articles related to 'intelligence' [WSHP99]. The shades of gray express distances between map nodes: If neighbored nodes contain similar document sets bright shades are used. Vice versa, dark shades reflect high dissimilarities between neighboring nodes of the grid. By clicking on the map a user can zoom into an area of interest until he or she finally selects a single grid node and receives a list of documents (with 10–15 articles in average).

To generate a document map WEBSOM applies the self-organizing map algorithm at two levels: First, a SOM is used to build a word category map based on short word contexts. These categories are then used to encode documents as word category histograms which are finally clustered by a second SOM. In more detail:

1. *Encode the documents as word category histograms.* Common and untrimmed term vectors used in standard retrieval are orders of magnitude too large to be

Table 5-3: Some technical details of WebSOM (cf. [KKLH96])

document space	number of documents	131,500 for training and mapping
	parts of documents used for indexing	full
	number of indexing terms (input vector dimension)	315
	weighting scheme	word category histograms
	dimensionality reduction	word category map
SOM	size of SOM (number of units)	768 (for training) $192 \times 256 = 49,152$ (by interpolation)
	training cycles	<i>unknown</i>
	map layers	multiple

handled efficiently with the neural network mapping. Furthermore, without using a thesaurus synonyms are treated as unrelated words. To achieve both a compact representation of documents and a similar encoding of synonyms the authors propose the use of a word category map, i.e. a ‘self-organizing semantic map’ introduced by Ritter and Kohonen [RiKo89] applied to word contexts [Honk97]. The idea of those maps is that words with similar short contexts have a similar meaning and thus should be mapped to a common category. Therefore, each non-stop word of the collection vocabulary is first encoded as an n -dimensional random vector x_i . This representation is then extended by $2n$ dimensions which express the estimated average value of previous and next words x_{i-1} and x_{i+1} , respectively, over the whole collection. The resulting $3n$ -dimensional vectors representing the words are then used to train a self-organizing map. As a result, words with similar contexts tend to occur in the same nodes of the grid, thus forming word categories. Finally, documents are encoded by using word category histograms rather than ‘isolated’ term histograms.

2. *Map the documents to a 2-dimensional grid.* A representative sample of the documents encoded as word category histograms is mapped to the grid using the SOM algorithm. The remaining documents are positioned on the scaled map (enlarged by interpolation) by taking advantage of the generalization feature of Kohonen’s model.
3. *Visualize the trained network.* To assign gray levels to grid nodes the unified distance matrix visualization method of Ultsch [UISi90, Ult93] (cf. chapter 4.2.3) is used. This method considers the distances of neighbored nodes. The gray scales reflect the relative average similarity of nodes in different observation areas. Thus, areas in the map representing dense parts of the document space are graphically displayed as bright gray shades, showing the clustering tendencies of the data set [KLHK98].

In a recent work the WEBSOM method has been applied to different types of collections [Lagus98], in detail: newsgroup articles, abstracts of scientific articles, patent abstracts, and news items. The processing time for the document maps varies from 1 hour (for 58 scientific abstracts) to 1 month (for 1,124,134 newsgroup articles). Table 5-3 summarizes some important technical parameters of a WEBSOM computation. Note that the information given relates to the computation of the document map, i.e. the self-organizing map used on the second level rather than the word category map on the preprocessing level.

5.6 Evaluations of Graphical Overviews for Searching Texts

Intuitively, the visual retrieval methods and tools – in particular the document landscape and document map approaches focused in this work – seem to be appealing and helpful for document retrieval and text-access. However, there are only few results which actually throw light on their usefulness for the intended application domains. In fact, for most of the systems and methods there seem to be no sound evaluations which give evidence, whether positive or negative, on their suitability. Marti Hearst [Hea99] states that “*evaluations that have been conducted so far provide negative evidence as to their usefulness*”, citing a work which suggests that non-expert users have difficulties in using graphical cluster presentations and would prefer textual representations (cf. [KLP96]). This section presents a brief literature review which aims at sketching results from evaluations performed so far. For this, it is interesting to not only look at results from document map evaluations but to extend the scope to other visual retrieval methods, too.

Regarding overview presentations of query result sets (cf. section 5.3.3), for the system ENVISION a usability study revealed that users can well understand the system and are satisfied by the presentation [NFH+96]. An evaluation of the frequency bar matrix display for query terms in result sets ([VeHe97], section 5.3.3) indicates its potential value but all in all provides mixed results [VeBe96]. For document similarity presentations in query result sets, Allan and Leuski studied the influence of the number of display dimensions on the effectiveness to which relevant documents can be isolated: Using structural measures instead of performing a user experiment, [LeAl99] has shown that the precision increases with the number of dimensions used (1D, 2D, 3D). An experiment involving users [AlLe00] revealed that the 3D presentation indeed performs best, but that it also clearly increases the cognitive load compared to the 2D presentation.

Though these evaluations provide some insight into aspects of methods for visualizing search results, they do not show their principal superiority over textually structured result sets alone. In [SVM+99] a comparative evaluation of text, 2D and 3D interfaces for visualizing search results which present clusters of documents and information on cluster density is discussed. It turns out that user performance with graphical interfaces depends on experience, and that overall the text interface showed the fastest response time. Furthermore, the authors conclude that the utility of graphical interfaces depends on their ability to reduce mental workload, and that an effective reduction depends on the right combination of task, user and interface. Morse et al. [MLKO98] compared five types of interface representations, including ordered text, ordered icons, a table format, a graph format and a spring-based visualization in order to assess their relative utility for retrieval tasks. The authors observed a significant difference between performance of users across the different interfaces. Though text interface and icon list yielded the highest performance, user preference clearly indicates that the text format was the least desirable and visual formats were preferred.

There are also some results regarding the graphical presentation of entire text collections: For analyzing a hierarchically organized text corpus, Shneiderman et al. examined the usability of a graphical system which presents hierarchies in terms of point clusters that are arranged in a grid [SFRG99]. It turned out that users achieve good results for analysis tasks like searching for clusters and assessing their sizes. Stasko et al. [SCGM00] compared different kinds of hierarchy presentations and their influence on the quality of analysis results and found that the style of presentation has a severe effect on effectiveness and user preference.

Turning to the document map approaches, Lin compared three different map presentations – his automatic category map approach and two human-generated maps – in order to study the usefulness of map displays for information seeking ([Lin95], cf. section 5.5.1). He found that, beyond significant differences in the effectiveness of using these presentations, the studied map displays generally help to learn and memorize structures of the underlying collection which speeds up the response time in subsequent searches. Users are supported to make judgments on the relevance of selected locations for the requested information and to find areas in the display that may be related to the query. Chen et al. compared their category map approach to the hierarchical catalogue organization of ‘Yahoo!’ for a broad browsing task [CHSS98]: Test subjects were asked to explore the information space without a specific goal, just trying to find a document of interest. From two groups, one started browsing with the category map, the other with ‘Yahoo!’. Whereas in both cases most test persons found an interesting document, only very few of the people who browsed with ‘Yahoo!’ were able to re-locate their document using the map while many of the map users could identify the location of their document in ‘Yahoo!’ in a following session. Though this may be due in part to the inconsistent mental models of both interfaces, it also shows that such a category map approach is not suitable for goal-directed searches (like finding documents again).

5.7 Discussion in the Context of Text Corpus Analysis

Exploring a collection of documents and understanding its structure of contents requires the existence of semantic search paths – or to follow Lauren B. Doyle: of ‘familiar conceptual grooves’. In this chapter non-visual as well as visual methods for providing the user with a certain access to a corpus of documents have been reviewed. Categorization, dynamic text based clustering and hypertext linking and structuring can improve the browsing capability of a searcher. But when it comes to intuitively conveying a picture of the whole corpus and presenting an overview of the collection’s similarity structure graphical methods seem to be more suitable – though a thorough evaluation is missing. Visualization in the context of document retrieval and management can be related to different aspects of documents, reaching from the topical flow of single documents to the display of certain similarity structures of complete collections. The emphasis of the considerations was put on document maps which focus on the display of a collection’s overall topical structure. The most important characteristics of the different approaches have been presented in the corresponding sections and some differences have already been pointed out explicitly.

This section summarizes these approaches and presents a deeper reflection on features which are important especially in the context of corpus analysis tasks for managing the knowledge contained in specialized document collections. Following a summary of the original application areas of the presented approaches, the methods and systems are compared against each other. Features to be discussed include the visualization principle applied, the degree of granularity of the display, the flexibility of the method with respect to the document compari-

Table 5-4: Overview of important features of the discussed approaches

method / system feature	approaches based on numerical scaling				SOM-based approaches			
	BEAD	GALAXIES/THEMESCAPE	VXINSIGHT	STARLIGHT/TRUST	Category Maps	Software Component Map	Hierarchical Classification Map	WEBSOM
individual documents visible?	✓	✓	✓	✓	—	✓	—	—
variable document comparison module?	—	—	✓	—	—	—	—	—
user influence on document comparison / adaptability?	—	—	✓	✓	(✓)	—	—	—
possibility to incorporate scenario related background knowledge?	—	—	—	—	—	—	—	—
cluster labeling?	—	(✓)	—	✓	✓	—	✓	✓
application-oriented evaluation?	—	—	—	—	✓	—	—	—

son module, as well as adaptability and user influence on the comparison. Table 5-4 gives an overview of the issues discussed in the following.

Regarding the **application areas** of the approaches presented in sections 5.4 and 5.5, most systems and methods aim at providing an interface for retrieving documents or browsing through collections. BEAD has been applied to a relatively small collection of bibliographic data consisting of abstracts of HCI proceeding papers, category maps were used to build a ‘table of content’ of Web pages. WEBSOM has been designed for browsing newsgroup articles – which is the focus of the work – and was marginally applied to small abstract collections of scientific papers and patents. KNOWLEDGE GARDEN as a groupware tool visually clusters manually collected and indexed WWW documents. It is not intended to serve as a means for visualizing the structure of a corpus in the first place and should thus be excluded from further discussions. GALAXIES/THEMESCAPE, VXINSIGHT and STARLIGHT/TRUST have been designed for ‘document mining’, i.e. database exploration in the context of strategic knowledge management. That means that these tools are either part of a more general system suite (cf. SPIRE’s GALAXIES/THEMESCAPE, section 5.4.2) or provide more flexibility and interactive features than the pure browsing tools.

Based on the **visualization principle** the approaches can be assigned to two groups. The group of work presented in section 5.4 (excluding KNOWLEDGE GARDEN) calculates a document space for visualization by using certain numerical scaling techniques (multidimensional scaling: STARLIGHT, GALAXIES/THEMESCAPE; force directed placement: BEAD, VXINSIGHT). Roughly spoken, based on a proximity measure between texts – adopted from information retrieval – for each document a point in a 2D or 3D metric space is calculated so that the distances between the points approximates the similarity of the corresponding documents. This space can be visualized directly by using a simple scatter plot or a more elaborated landscape metaphor (3D point clouds: BEAD, STARLIGHT; 2D clouds: GALAXIES; mountain terrain/landscape: VXINSIGHT, later version of BEAD). The scaling techniques used try to opti-

mize the *distances* between the documents with respect to the given proximity measure. Instead of displaying *absolute metric distances* between documents the self-organizing document map approaches (section 5.5) try to preserve the *topological structure* of the document space as good as possible. This means that the underlying visualization principle is to reflect the neighborhood relationships rather than the absolute distances, leading to a more compact graphical representation of the document space.

Another criterion for visualization regards the **granularity of the display**. Here, the question is whether only information on a meta-level is provided (i.e. only the relationship of categories or the space's density is shown) or whether the user can gain insight into the relationships of single documents (cf. criterion 'focused relationship' in the task model, section 2.6.1). The approaches based on numerical scaling techniques are focused on the display of similarity between *individual* documents. In contrast, category maps present the associative structure of document collections in terms of neighborhood of categories. Between these extremes WEBSOM presents the density of a collection by providing a 'landscape of documents' where each grid unit contains a set of very closely related texts. Thus, WEBSOM performs an *aggregation* of information. A certain notion of density is also conveyed by ESA's hierarchical classification map which displays the matching frequency for the single units. In the (very small) software component map (section 5.5.2) each document is usually assigned to a single node which displays the matching document's name. However, the user gets only a very coarse idea of the relatedness of single documents.

For supporting the different characteristics of various document collections and the large spectrum of possible analysis tasks it is desirable to be able to choose an **adequate indexing and comparison model** for the document collection if available (cf. chapter 3). The document comparison module determines the similarity of texts and, thus, forms the basis for visualizing the structure of document collections. In principle, the approaches based on numerical scaling techniques only require a (dis-)similarity measure for text objects, independent from the formal representation of documents, though all approaches discussed in this section use a fixed term vector representation. TRUST even depends on a term vector representation of documents due to its concept space visualization. Only VXINSIGHT provides a selection of pre-defined similarity measures (similarity based on link structures or shared keywords, for example). The labeling technique of the category maps depends on term vector representations, and WEBSOM uses a SOM-based word category histogram encoding for documents in its overall architecture.

An interesting question in the context of text-mining and knowledge management is whether the approaches allow the user to **influence the way the documents are arranged** in the display, possibly **based on background knowledge**. VXINSIGHT allows the user to select a document similarity function from a pre-defined set as a basis for graphically presenting the similarity structure (see also section 5.3.1). But are there stronger means of realizing adaptability that allow to define a personal 'view' on a collection of documents? There are two ways to realize such a view: The first way follows a certain 'principle of exclusivity', i.e. only the user-defined interest is regarded for generating a document map. In STARLIGHT/TRUST the user can select concept axes and thus define a 'concept sub-space' in which documents are arranged. This sub-space exclusively reflects the interest of the analyst. Systems which organize a sub-set of documents with respect to a special interest are often referred to as 'dimension systems' or 'reference point systems'. In contrast, the extended category map approach of Chen rather follows the 'principle of inclusivity' since it displays the entire document collection with respect to its structure and the special interest. Here, adaptability is realized by adjusting terms used for building categories, but at the same time, other, non-adjusted terms are regarded for map generation. In other words, document features which have not been defined

explicitly by the analyst are considered (or included), too. None of the approaches presented here allows to incorporate scenario-related background knowledge into the process of arranging documents.

What additional features should a document map system possess? Browsing through a collection and understanding the structure presented in the display requires that the visualization tool provides a **means for describing the clusters**. Regarding this feature STARLIGHT/TRUST, the category and the hierarchical classification map approaches as well as WEBSOM present the dominant key words of each document group, and THEMESCAPE labels the topics of the collections.

Understanding the usefulness of graphical overviews requires a thorough **application-oriented evaluation**. In fact, this is the weakest point of the research conducted so far. Most of the document map approaches aim at supporting a search in unrestricted document collections. The added value of the approaches over alternative search methods and text-access paradigms has rarely been examined. An exception in this context is the category map approach for which different evaluations have been conducted by two research groups (cf. section 5.5.1). For the rather text-mining-oriented approaches (which are partially commercialized) application-oriented evaluations are not available at all.

Regarding the **sizes of the target or test collections** – as far as information is available – BEAD and the software component map operate on small collections. This is suitable due to the narrow focus of the work (pre-selected bibliographic data or software descriptions, respectively). STARLIGHT/TRUST is used with Boeing internal data sets consisting of up to 40,000 documents. To allow quick operations and user interaction on this corpus it relies on a fixed document representation and a distance preserving graphical display. The target collections of category maps and WebSOM are large amounts of documents from the Internet. The SOM-architecture of the category map approaches is adapted to the problem of categorizing large collections, i.e. mapping of many documents to a low number of grid nodes. In fact, Chen's category maps have been tested with up to 10,000 documents. WEBSOM trains relatively small maps and scales them afterwards by interpolating the weights associated with each node in order to map large numbers of documents (up to 1,000,000, realized by using a 512-processor neuro-computer [KKLH96]). Table 5-5 gives an overview of the order of magnitude of the document collections under consideration. Comparable statements regarding the total computation time of the different approaches can hardly being given. If reported by the authors at all, different (sometimes incompletely described) hardware is used. Some authors only claim real-time interaction but there are no benchmarks for the often expensive computational effort for preprocessing the data.

Table 5-5: Order of magnitude (number of documents) of the test/target collections

BEAD (cf. section 5.4.1)	GALAXIES/ THEMESCAPE (cf. section 5.4.2)	VXINSIGHT (cf. section 5.4.3)	STARLIGHT/TRUST (cf. section 5.4.4)	Category Maps (cf. section 5.5.1)	Software Component Map (cf. section 5.5.2)	Hierarchical Classification Map (cf. section 5.5.2)	WEBSOM (cf. section 5.5.3)
100 – 300	<i>unknown</i>	<i>unknown</i>	≤ 40,000	100 – 10,000	< 100	1000	100 – 1,000,000

5.8 Motivation and Goals of the Own Approach

The evaluations of graphical overviews provide mixed results as to their usefulness (cf. section 5.6). In summary, graphical overview approaches may reduce the cognitive workload for certain analysis tasks, like examining retrieval results or deriving structural information. They have the potential to be more appealing to users than textual presentations. Document maps, in particular, seem to be inappropriate for goal-directed retrieval tasks or navigation but convey a pretty good picture of structural aspects of a document collection. Working effectively with such interfaces, however, requires a certain experience and is not suitable for naïve, occasional users. Rather, a fruitful application of document maps requires a well-tuned combination of domain, scope of users, as well as method and tool design.

Starting point of this work is the hypothesis that document maps may well be of value for certain text-access tasks which arise in managing the knowledge contained in thematically specialized corporate document collections (cf. chapter 2). Core of their intended application is not goal-directed retrieval or browsing, but rather the analysis of a collection's structure for expert users, i.e. the support of tasks which involve pattern discovery of some kind, such as grouping related documents, studying the topical document distribution, examining content-based relationships between individual documents, or mining valuable textual assets. In such a context document maps may increase the effectiveness of performing complex corpus analysis tasks in knowledge management and reduce the cognitive workload for the analyst. This thesis aims at designing and evaluating a document map approach which allows a detailed and fine-granular analysis of the structure of specialized, moderately-sized text collections. This scope, however, requires a powerful technical design which goes beyond approaches proposed so far. The next sections discuss the features and research questions which arise in the context of the intended application domain.

5.8.1 Visualization Principle and its Degree of Granularity

The first requirement for the proposed document map approach for corpus analysis in knowledge management concerns the degree of granularity and the visualization principle of the graphical presentation. According to the task model (chapter 2) practically relevant corpus analysis tasks are performed on both, an overview level where only a coarse grouping of documents is relevant as well as on a detailed level where differences and relationships of individual documents are of particular importance. In many tasks fine-granular document-to-document relationships are interesting (cf. criterion 'focused relationship' in the task model in section 2.6.1 and examples of analysis tasks in sections 1.1 and 2.6.2). Thus, for a fine-granular understanding of the structure and relationships of documents within a specialized corpus individual documents should be visible.

Which visualization principle should be pursued? Usually an analyst is not interested in 'measuring' the absolute metric distance of text representatives in the graphical display (what is the meaning of a numeric degree of relatedness measured by a document analysis method, anyway?) but rather cares for the overall topological structure of the document space, i.e. the grouping of more or less related objects and the grouping of text clusters. In other words, the visualization technique should powerfully reflect the cluster structure of the document space. This is an application where the self-organizing document map approaches are promising (cf. sections 4.2.2 and 5.5) since they present neighborhood relationships of texts instead of document distances and are thus capable of yielding an intuitive and compact representation of the similarity structure of a text corpus.

5.8.2 A Modular Approach for Structuring Specialized Text Collections

Which method is suitable for computing document relatedness in the application domain ‘managing the knowledge contained in specialized text collections’? The range of relevant documents includes semi-structured texts, like requirement definitions, as well as knowledge-intensive documents, e.g. patents or technical manuals. Chapter 3 has shown the variety of text representation methods and measures of document relatedness that can be found in text retrieval, information extraction, textual case-based reasoning and related areas. The spectrum of representation forms reaches from simple term vectors to structured domain-specific feature representations with rich semantics. Measures for text similarity or dissimilarity include lexical-based measures, measures based on explicitly given links (citation or hypertext links), or measures that incorporate domain theory. In particular for specialized application domains there are domain-tailored methods for representing and comparing texts, e.g. legal texts (3.3.1), medical documents (3.3.2, 3.3.4), scientific and technical literature (3.2.1, 3.3.2, 3.3.3).

The scope of this work is computing and graphically presenting the similarity structure of a specialized text corpus (e.g. in enterprises or for technical and scientific applications) in order to support corpus analysis tasks. Obviously, the analysis of documents regarding their semantic similarity is the key concept for generating a document map (even more important than the visualization itself). But what kind of document similarity or relatedness shall be displayed? The question of which document analysis method fits best cannot be answered a priori due to the diverse nature of relevant collections and tasks, and the various strengths, weaknesses and foci of different document analysis techniques. In the simplest case the analyst may be satisfied by a simple kind of overall similarity based on keyword distribution. But maybe he needs to analyze a collection of technical documents regarding deep technical details and relationships. Or the analyst is a researcher who wants to study a collection of scientific papers regarding their citation structure. The simple consequence is that the document map approach should not be pinned down to a particular document analysis module. In order to have a suitable and reliable basis for visualizing the similarity structure of a text corpus the method for representing documents and for determining their similarity should be exchangeable so that the special needs of the collection under consideration and the task to be performed can be met (see also [Lenz98b]).

5.8.3 Application-Oriented Evaluation

The main hypothesis of this work is that document maps – though they may be rather inappropriate for classic browsing and searching (cf. section 5.6) – can effectively support corpus analysis tasks in knowledge management. Thus, a profound application-oriented evaluation is imperative for understanding the task-adequacy of document maps in the proposed domain. Otherwise, only limited additional insight could be gained regarding the issue of effective graphical overview methods for corpus analysis and text-access – no matter how innovative the developed methods are. The evaluation has to be guided by real-world tasks, supported by the task model developed in chapter 2 which provides a detailed taxonomy for characterizing typical and practically relevant analysis tasks for text corpora in the context of knowledge management. The following questions need to be addressed: How can the proposed document map approach be productively applied for solving real-world tasks? Does the approach yield a qualitative/quantitative added value? Which facets of analysis tasks are effectively supported with respect to the task model? If the general approach turns out to be valuable, where are starting points for improvement with respect to the application domain?

5.8.4 Extension: An Adaptable Document Map Approach

An additional requirement that should be met in the context of the proposed application domain is the adaptability of the document map approach in the following sense: In a knowledge management context different analysts may be interested in different aspects of the document collection, though they all share a common agreement about global criteria for document relatedness. There may be different personal weightings of document relationships which reflect the individual interest from analysts for a particular analysis task. For instance, the supervising manager of a project will often be interested in other aspects of a collection of brainstorming results than the developer is. An engineer might want to stress some technical features described in a collection of technical documents whereas a business administrator might be more interested in marketing-relevant information. The fact that different people may be interested in different facets of the document collection leads to the concept of ‘views’ – or to the demand for an adaptable document map approach.

Thus, the method to be developed should provide a means for adaptability, i.e. the analyst should be able to influence the map generation process by incorporating scenario-related background knowledge, if necessary. Following the terminology used in section 5.7 there are two ways to realize such a view: The ‘principle of exclusivity’ assumes that the analyst has a rather clear understanding of relationships to be examined and wants the graphical presentation to reflect solely the similarity of documents regarding a clearly defined subset of features. In contrast, the ‘principle of inclusivity’ is particularly useful when the analysis is less focused in content and excluding potentially helpful semantic connections would impede the analysis process. Though both ways are important and interesting in the context of corpus analysis and text mining, this work concentrates on the second principle, i.e. on exploration tasks where a view should consider both, a general understanding of document relatedness and a personal interest. Here, rather than being exclusive, the personal perspective should only form a certain *bias* regarding the structures displayed by the map: *In addition* to a priori given document relationships it should determine a degree of relatedness between documents and smoothly incorporate this information into an existing document map.

5.8.5 Summary of the Discussion and Overview of Contributions

Summing up, for corpus analysis tasks in specialized document collections a map approach seems to be appropriate which presents the similarity structure of individual documents based on a ‘compact’ self-organizing mapping and which allows the use of adequate document comparison techniques for the collection under examination and the task at hand. An interactive system should provide flexible means for analyzing the corpus and interacting with the map display. Furthermore, an interesting additional feature is a component for realizing adaptability. As the presentation of document landscape and document map systems in this chapter has shown, an approach which combines these features is missing. Another important research issue which needs to be addressed (beyond developing pure technology) is to gain a deeper understanding of the value added by graphical overview tools for the intended application domain. Indeed, this is a scarcely considered issue. Especially for the application domain of corpus analysis there are no results which provide evidence regarding the usefulness of document map approaches.

Therefore, in this thesis the concept of document maps is studied intensively – from design to evaluation – in the context of exploring and analyzing specialized document collections. In particular, the following contributions are made:

- development of a modular framework for computing document maps which allows the flexible integration of collection-tailored methods for semantically comparing documents (chapter 6),
- development of an interactive document map system which comprises powerful tools for analyzing the text collection under consideration, tightly coupled with the map display (chapter 7),
- application-oriented evaluation of the method in industrial and scientific case studies in order to work out productive ways to apply the tool and to understand and assess its value (chapter 8),
- a task-oriented quantitative evaluation of the performance of the proposed document map approach which allows to generalize results regarding its usefulness on an empirical basis (chapter 9),
- and finally the additional development of a means for incorporating adaptability into the document map framework which allows a user-driven refinement of detected structures by applying scenario related background knowledge, allowing the analyst to form a bias towards his special interests (chapter 10).

6 A Basic Framework for Document Maps

This chapter introduces the basic framework for generating document maps and presents its realization in this work. Basically, the framework composes well-known methods from explorative data analysis, in particular multidimensional scaling, machine learning and data mining in order to support effective information access for knowledge management tasks. The added value of this composition will be a flexible yet powerful model for visualizing the semantic structure of a collection of documents. The flexibility of the model concerns the exchangeability of the core-element of semantic structuring: the method for comparing documents with each other. The chosen design of the framework will make it possible to select a suitable model for document analysis with respect to the special requirements of different application contexts. The idea of the framework, its design and realization have been published in [BST98, BSJ00a, BSJ00b].

This chapter is organized as follows: First, the overall design of the basic framework will be introduced and discussed (section 6.1). The main part of this chapter is concerned with presenting the realization of the framework in this work (section 6.2). The metaphor of resulting document maps and a discussion of their interpretation will be presented in section 6.3. After that, the parameters of the framework will be summarized and information about their proper setting will be given (section 6.4). Finally, first experiments with the approach will be sketched (section 6.5).

6.1 Design of the Basic Framework: Overall Architecture

The framework presented in this chapter is the basis for the proposed document map approach. Though it will be extended in chapter 10 by additional features its basic design already results in a high flexibility. This is because the framework is a strictly modular one with only loosely coupled components. Some of the components are fixed, others are exchangeable. The motivation for designing such a modular approach can be found in chapter 5.8.2.

Figure 6-1 sketches the modular approach for generating document maps. In the following, the design of this framework will be discussed in detail. The concrete realization of the approach is presented in section 6.2.

6.1.1 Variable Document Analysis Module

The input to the model is a collection of textual documents. Given the requirement of exchangeability of the document comparison method, what is the common denominator of different text matching techniques? Surely, it is not the way the documents are formally represented by a method since numerical as well as symbolic document representation can be found. Neither it is the way documents are matched against each other since this process clearly depends on the formal representation. In fact, the only thing one can expect from a document comparison technique is that it calculates a certain degree of similarity or dissimilarity of each pair of documents. Since we strive for a spatial representation of document

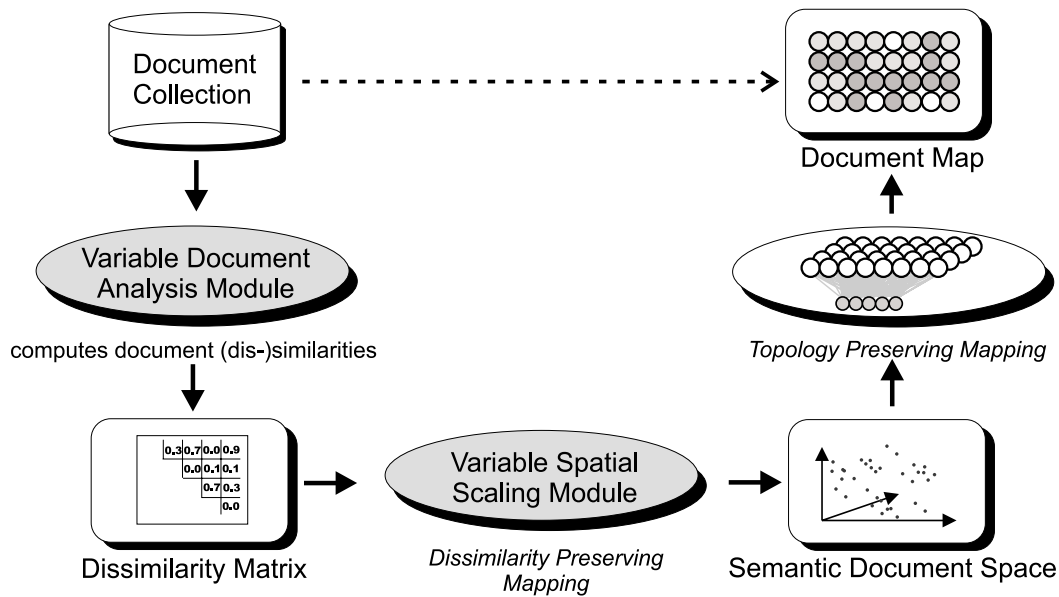


Figure 6-1: Architecture of the basic framework

similarity it is necessary to interpret similarity as distance information: The more similar two documents are, the closer they lie together in a suitable space. This space will be a metrical one, so metrical distance information will be needed to encode the documents' similarity as spatial information. However, it would be too hard a constraint to expect the document analysis module to yield only metrical distance data (i.e. document distances in a geometrical sense). Thus the requirements are relaxed and the document analysis module is expected to compute only a dissimilarity value for each pair of documents. More details on the document analysis module, the general conditions it has to satisfy and its prototypical realization in this work can be found in section 6.2.1.

As the only interface expected for the following processing steps is a matrix of dissimilarity values for each pair of texts the document analysis module is exchangeable. The internal document representation of this component is not used for any further step. Consequently, this module can be chosen with respect to the characteristics of the corpus and the requirements of the application domain – which is an important contribution of this architecture.

6.1.2 Variable Spatial Scaling Module

The next step for generating a document map is to construct a spatial representation of the documents based on the respective dissimilarity data. More precisely, a representative for each documents in a multidimensional space is calculated so that the (metric) distance between each pair of documents reflects the dissimilarity information – which is calculated by the document analysis module – as faithful as possible. This spatial representation is called 'semantic document space' in the following. Since the calculated dissimilarity information of pairs of documents may be non-metrical (which means in particular that the triangular inequality may be violated, cf. section 6.2.1.1) the spatial scaling module has to deal with both, metrical and non-metrical input data. This is exactly the task of techniques from multidimensional scaling and related approaches (cf. chapter 4). In this work a prototypical scaling module is applied which shows good results for metrical as well non-metrical dissimilarity data. It is a matter of accepted computational effort and desired accuracy whether other approaches would be applied instead in an alternative realization of the framework. Section 6.2.2 describes the chosen scaling method and its application to non-metrical dissimilarity data in detail.

It is important to note that though many document comparison and retrieval approaches directly represent documents in a vector space (e.g. the vector space model, cf. section 6.2.1.3) this should not be required in general according to the discussion in section 5.8.2. If a symbolic document encoding is given a spatial document representation exists only after the spatial scaling module has been applied. If, however, the document analysis module itself represents documents as vectors the spatial scaling of dissimilarity data can be seen as a dimension reduction technique.

The space constructed by the scaling module will not be visualized directly (in contrast to other approaches discussed in chapter 5). If that would be desired the semantic document space could only consist of two or three dimensions. But when the documents' similarity structure is encoded in only a few dimensions in general a heavy distortion of the dissimilarity values would have to be taken into account. Rather than displaying absolute distances the cluster structure induced by the dissimilarity information is visualized in this approach (cf. section 5.8.1). Since this structure is extracted by the separate topology preserving mapping module it is possible to use as many dimensions as necessary to preserve the dissimilarity information as faithful as possible (though there is a trade-off between accuracy and computation time for the final step, cf. section 6.1.3).

Some additional notes are appropriate here. First, the topology preserving mapping module in the final step of the modular pipeline measures the distance in the semantic document space by the Euclidean distance function. Consequently, the selected spatial scaling module should use this notion of distance, too. Alternatively, the distance function in the topology preserving mapping module should be exchanged appropriately. Second, the distance matrix is only a logical construct which does not necessarily have to be computed in full a priori. Depending on the design of the used spatial scaling module the dissimilarity values can be computed 'on the fly' when required.

6.1.3 Topology Preserving Mapping and Visualization Module

At this point a semantic document space has been generated which reflects the document similarities in its topology. Since in this work the cluster structure of the document space is regarded to be more important than metrical distances between documents a topology preserving mapping method is applied. More precisely, the topological structure of the semantic document space is mapped to a two-dimensional grid using the neural network model of self-organizing feature maps (cf. section 6.2.3). This neural network orders the input patterns from the semantic document space – i.e. the numerical vectors which represent the documents – according to their distance in two dimensions without losing too much topological information. The arrangement is realized in a self-organizing and unsupervised manner. A suitable visualization technique finally extracts the relevant topological information and displays the desired map.

Crucial for the performance of the self-organizing map (i.e. the computing time for the network's training phase) is the number of dimensions of the semantic document space. In practice this implies a trade-off between, on the one hand, the accuracy with which the dissimilarity information – as computed by the document analysis module – is preserved in the semantic document space (which depends on the number of dimensions used for the spatial scaling) and, on the other hand, accepted computation time for training the neural network.

6.2 Realization of the Document Map Approach

Having discussed the general design of the modular approach this section presents the realization of the framework in the present work. In this realization a concrete prototypical document analysis method is used – which does not undermine its exchangeability – and a suitable spatial scaling technique is applied. Each subsection now presents the chosen and partially adapted methods in detail and discusses the interfaces between the respective modules.

6.2.1 Document Analysis Module

The document analysis module takes the given document collection as input, computes a formal document representation and determines a degree of dissimilarity for each pair of documents. As the discussion in chapter 2 has shown there are many ways to assess the similarity or dissimilarity of textual documents. An important feature of the framework proposed in this work is that the document analysis module is exchangeable. This is realized by loosely coupling the document analysis module to the remaining computation steps: The only interface expected for the further calculation of the document map is a matrix of dissimilarity values for each pair of documents.

In the next subsection some basic properties which a document analysis module has to satisfy are discussed. After that the ‘default’ document analysis module which is used throughout this work will be presented. Finally, as the interface to the spatial scaling module is a dissimilarity function rather than a similarity function, the conversion from similarity to dissimilarity values will be discussed.

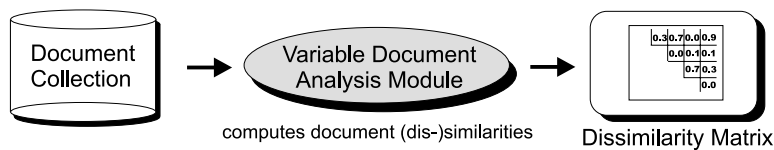


Figure 6-2: Context of the document analysis module within the basic framework

6.2.1.1 Generic External Specification for Document Analysis Modules

As the document analysis module is only loosely coupled with the rest of the proposed framework it is possible to apply a variety of document comparison and document retrieval models. However, there are some conditions, though straightforward, which have to be defined properly in order to have a solid basis for the presentation of the framework.

Which document analysis modules fit into the framework? The main components of a document analysis model (whether for document retrieval, case-based reasoning or other related purposes) are (a) a document representation scheme (also called indexing scheme), (b) a query representation scheme in the case of retrieval models, and (c) a retrieval function which calculates a retrieval value for a document with respect to a given query or a matching function that computes the relatedness of a pair of documents. Regarding query-oriented retrieval models, in general the document analysis module can be realized by all approaches which allow for document-to-document comparisons, i.e. the structure of both the document and the query representation scheme is equal. For example in case-based reasoning the query is usually a case description itself which is matched against other case descriptions. More specifically, a suitable model for document matching consists of

- (a) an indexing method which transforms a given full-text document d^* into a specific formal representation d . The set of all formal document representations $\{d_1, \dots, d_n\}$ is denoted as D ;

- (b) a matching function $M: D \times D \rightarrow \mathbb{R}_0^+$ which determines the degree of (dis-)similarity for each pair (d_a, d_b) of document representations.

Note the difference between the notions ‘document’ and ‘formal document representation’: The term ‘document’ refers to the human-readable text object whereas a formal document representation is encoded by means of a suitable ‘knowledge representation language’, e.g. lists of weighted key-terms, descriptors as defined by document description languages, logic-based representations, etc. (cf. chapter ?). Having said that, in the following the term ‘document’ will be used for both, its ‘natural’ as well as its formal representation because the meaning will be clear from context.

The document matching function $M: D \times D \rightarrow \mathbb{R}_0^+$ can be realized either by a measure of *similarity*, i.e. $\sigma: D \times D \rightarrow [0, s_{max}] \subset \mathbb{R}_0^+$ or a measure of *dissimilarity*, i.e. $\delta: D \times D \rightarrow \mathbb{R}_0^+$.

A measure of similarity is formally defined by a function $\sigma: D \times D \rightarrow [0, s_{max}] \subset \mathbb{R}_0^+$ for which the following properties hold for all $d_a, d_b \in D$:

$$(S1) \quad \sigma(d_a, d_a) = s_{max},$$

$$(S2) \quad \sigma(d_a, d_b) = \sigma(d_b, d_a).$$

A measure of dissimilarity is realized by a certain ‘distance’ function δ which determines the degree to which documents d_a and d_b are dissimilar. In other words, high similarity of documents corresponds to low dissimilarity and vice versa. Depending on its properties δ can be a metric, a pseudo-metric or a simple dissimilarity measure. More specifically, a function $\delta: D \times D \rightarrow \mathbb{R}_0^+$ is called *metric* on D if and only if for all $d_a, d_b, d_c \in D$ the following properties hold:

$$(D1) \quad \delta(d_a, d_b) = 0 \Leftrightarrow d_a = d_b,$$

$$(D2) \quad \delta(d_a, d_b) = \delta(d_b, d_a),$$

$$(D3) \quad \delta(d_a, d_b) \leq \delta(d_a, d_c) + \delta(d_c, d_b).$$

Condition (D1) can be relaxed to (D1*) so that $\delta(d_a, d_b) = 0$ for $d_a \neq d_b$ is allowed, too. In this case δ is called *pseudo-metric*. If δ satisfies only conditions (D1*) and (D2), the function is called *dissimilarity measure*.

Before the default document analysis module as used in this work will be presented, some notes on the definitions made in this section are appropriate.

6.2.1.2 Some Notes on the General Conditions and their Effect on the Framework

It is helpful to discuss the properties from above in order to get a more solid feeling for the problems that may arise when designing or choosing a suitable document analysis module. Though mathematically straightforward there are some practical and philosophical problems coming along with these definitions. Most important, in many application areas the concept of symmetry for notions of dissimilarity or similarity (S2, D2) and the triangular inequality (D3) do not hold (cf. [Leng96, Davi83]).

Concerning the concept of symmetry, most people would regard a sub-concept to be more similar to the super-concept than vice versa. Applied to the problem of comparing documents this phenomenon may arise when a more general document is compared with a more specific one. However, in information retrieval the concept of symmetry is quite commonly assumed (at least for document-to-document comparisons). For document maps it is important to assume symmetry of document comparisons, since the visualization concept pursued here is not able to adequately represent asymmetric document-to-document relationships.

Another philosophical problem concerns the triangular inequality: A document about air-planes may be regarded as similar to a document about space ships because both documents are about flight vehicles. The space ship document may be judged similar to a document about asteroids because both documents deal with issues relevant in space travels. However, why should the airline document be similar to the asteroid document? Consequently, a reasonable measure of document dissimilarity may judge the plane and the asteroid texts as highly dissimilar and thus spatially far apart whereas the dissimilarity for the space ship text to both, asteroids and planes, is very low and thus spatially close. As a consequence this conception of measurement may violate the triangular inequality. The document maps proposed in this work, however, are based on the visualization of a high-dimensional *metrical* space, i.e. the triangular inequality is assumed implicitly. In cases where the triangular inequality does not hold a certain distortion of the original dissimilarity values is inevitable. This is a classic trade-off problem between a certain information loss on the one hand and a gain of ‘accessibility’ to a complex information space on the other. In spite of this problem document maps promise a high value for practical applications. Note that document maps do not aim to *classify* a given set of documents but visualize its similarity structure. Ultimately, coping with the triangular inequality problem is a matter of interpreting the map.

As a final note consider properties (D1) and (D1*) and the notion of ‘pseudo-metrics’: A pseudo-metric may be used if one is only interested in comparing certain key-features of two documents d_a and d_b . The considered documents may share these features whereas other features which distinguish d_a and d_b are not of interest, thus the dissimilarity of d_a and d_b in this regard is zero. From a technical point of view this should not produce any problems. Instead of considering δ to be a pseudo-metric on the original document representations d_i it would be possible to regard δ to be a metric on the considered subset of features.

6.2.1.3 The Vector Space Model as an Example of a Document Analysis Module

In this section the default document analysis module is presented: The vector space model [Salt71] is a simple and well-known retrieval model which allows document-to-document comparisons. In that approach both documents and queries are represented as numerical term vectors. A variety of different retrieval functions – all based on vector algebra – has been proposed in literature. The common assumption is that document vectors (or document and query vectors) that are close in space or ‘point’ to similar directions correspond to similar documents (or matching queries, respectively). As the vector space model is the default document analysis model in this work, and, furthermore, some system functions of the developed prototype system (cf. chapter 7) use the vector space method (which will not undermine the exchangeability of the document analysis module), a short discussion of the approach will be given in this section.

Given a collection D^* of n text documents d_1^*, \dots, d_n^* , each document is processed by the following procedure:

- Break the text into words and remove semantically irrelevant words like conjunctions, articles and other insignificant terms (so-called stop words) from the documents using a stop word list.
- Reduce the remaining words to their stem using a computer-linguistic stemming procedure (cf. section 6.2.1.4).

The resulting set $T = \{t_1, \dots, t_N\}$ of all relevant and stemmed terms of the collection serves as the indexing vocabulary. Each document d_i^* can now be represented by a term vector $d_i =_{\text{def}} (d_{i1}, \dots, d_{iN})$, $d_{ij} \in \mathbb{R}_0^+$, where each component d_{ij} corresponds to term $t_j \in T$ and is called the term weight of term t_j . The way of how to determine the weight for a term in a given docu-

ment can be defined individually in each realization of the method. According to [SaBu88] each term weight should be based on a local component $local(i,j)$ which depends on information from the i -th document about term t_j , a global component $global(j)$ which depends on information on term t_j from the whole collection D^* of documents, and a document-dependent normalization component $norm(i)$, i.e.

$$d_{ij} = local(i,j) \cdot global(j) \cdot norm(i). \quad (6-1)$$

In this work two weighting schemes are used: normalized term frequency and normalized $tf \times idf$ weights. In both cases the local component simply counts the number of occurrences tf_{ij} of term t_j in document d_i^* , i.e. $local(i,j) =_{def} tf_{ij}$. In the normalized term frequency scheme there is no global information about t_j , i.e. $global(j) =_{def} 1$. In contrast, in the $tf \times idf$ weighting scheme the global component is called the inverse document frequency of t_j in collection D^* , defined by $global(j) =_{def} \log(|D^*|/n_j)$ where n_j denotes the number of documents containing t_j . Note that most of the term vectors' weight components typically are zero-entries. In order to compensate the influence of the documents' lengths on the weights in both cases the term vectors are normalized to unit length, i.e. $norm(i) =_{def} \|d'_i\|^{-1}$ where $d'_i =_{def} (d'_{i1}, \dots, d'_{iN})$ and $d'_{ij} =_{def} local(i,j) \cdot global(j)$. The resulting set of formal document representations is denoted as $D = \{d_1, \dots, d_n\}$.

The (dis-)similarity of two given documents d_a^* and $d_b^* \in D^*$ can now be defined by a vector algebraic matching function which works on the corresponding term vectors d_a and $d_b \in D$. Many realizations for matching functions have been proposed in literature (cf. [SaBu88, Korf97]). Both, measures of similarity and measures of dissimilarity (distance) can be used. In this work the cosine measure of similarity and the Euclidean distance to compare pairs of documents are used. Each of these measures is based on a different perception of 'document similarity': Whereas the Euclidean distance approach follows the idea that spatially close term vectors correspond to similar documents (the lower the distance, the more similar the documents), the 'philosophy' of the cosine measure of similarity is that vectors pointing to similar directions are related to similar documents. The cosine measure of similarity is defined as the inner product of a pair of term vectors (provided that the term vectors are normalized to unit length), the Euclidean measure is a special case of a p -norm with $p = 2$:

$$\begin{aligned} \sigma_{cos}(d_a, d_b) &=_{def} d_a \cdot d_b = \sum_{j=1}^N d_{aj} \cdot d_{bj}, \\ \delta_E(d_a, d_b) &=_{def} \|d_a - d_b\|_E = \left[\sum_{j=1}^N (d_{aj} - d_{bj})^2 \right]^{1/2}. \end{aligned} \quad (6-2)$$

It is trivial to check that σ_{cos} satisfies condition (S1) and (S2) from section 6.2.1.1. The well-known Euclidean distance δ_E is, of course, a metric.

The vector space model in its basic form performs literal term matching. To cope with the variety of expressions for similar concepts in natural languages the usage of thesauri is recommendable. By doing this, each term t_j represents a set of semantically similar words instead of a word directly derived from a given document. In this work domain specific thesauri can be used for indexing documents using the vector space method (cf. chapter 7.3).

6.2.1.4 Excursus: Stemming

Stemming is a linguistic procedure which aims at normalizing words, i.e. reducing different word forms (inflection forms and derivation forms) to a common root form – called 'stem' – which thus represents a set of 'related words'. For example, consider the words 'compute', 'computes', 'computation' and 'computers' which can be reduced to their stem 'comput'.

In information retrieval stemming is used to improve the quality and performance of retrieval systems. However, the methods applied here are not able to perform the stemming task always correctly from a grammatical point of view. Rather, they are intended to yield a reasonable high precision in order to support practical retrieval tasks.

Stemming mostly concentrates on suffix stripping (cf. [Korf97]), i.e. the removal of grammatical endings (declension forms of nouns, conjugation forms of verbs) and morphological endings (which create different word types, e.g. the English ending *-ly* makes an adverb from an adjective).

There are different approaches for stemming:

- Linguistic algorithms apply rules for a stepwise reduction of a given word to its stem and are adequate for weakly inflected languages like English. Currently, the most prominent type of linguistic stemming algorithms is the class of *iterative affix removal stemmers* [Korf97].
- Lexical approaches are based on full-form dictionaries which contain the base form for all inflection and derivation forms. Moreover, these dictionaries are often used for word decomposition. Lexical approaches are more suitable for strongly inflected languages like German.

The Porter stemming algorithm [Port80] is a widely accepted iterative affix removal stemmer which yields a sufficient precision for many applications. A detailed description of this algorithm would go far beyond the scope of this section. However, presented on a general level, the differences between various iterative affix removal stemmers are not obvious since the core of each algorithm is the choice and the order of rules to be applied.

To sketch the basic procedure of the Porter stemmer in short: The algorithm defines various rules of the form (*condition*) $S1 \rightarrow S2$, meaning that if a given word ends with the suffix $S1$ and the string preceding $S1$ satisfies the condition, $S1$ is replaced by $S2$. There are different groups of rules, and from each group the rule with the longest match for $S1$ is applied. A given term successively passes the defined steps of suffix removal: The first groups of rules remove the grammatical endings. After that the morphological endings are eliminated. Finally, remaining ‘tidying up’ tasks like replacing double letters in endings by single letters are performed. As a result of the grouping complex suffixes are removed bit by bit, e.g. oscillators \rightarrow oscillator \rightarrow oscillate \rightarrow oscill \rightarrow oscil.

The present work uses an implementation of the Porter algorithm by Frakes and Cox [FrCo91] for English and the implementation of a Porter-style stemming algorithm for German designed by Stieger [Stie93].

6.2.1.5 Converting Document Similarity Values into Dissimilarity Information

As discussed in section 6.2.1.1 the document matching function can be realized by a similarity measure $\sigma: D \times D \rightarrow [0, s_{max}]$ or a (possibly metrical or pseudo-metrical) dissimilarity function $\delta: D \times D \rightarrow \mathbb{R}_0^+$. However, the spatial scaling module’s input is a dissimilarity function $\delta: D \times D \rightarrow \mathbb{R}_0^+$ which provides information about the proximity which documents d_i and d_j should have in the semantic document space. This proximity information reflects the documents’ similarity: The more similar documents are, the closer are their distances (or proximities). Consequently, similarity measures σ have to be transformed into suitable dissimilarity measures δ_σ . This section proposes different transformation functions for this step.

First, it is necessary to define some general conditions which a suitable transformation function should possess in order to harmonize with the intuitive understanding of ‘reflecting simi-

larity by distance'. Consequently, a *similarity-to-dissimilarity transformation* is defined as a function $\tau : [0, s_{\max}] \rightarrow \mathbb{R}_0^+$ with the following properties:

- (i) $\tau(s_{\max}) = 0$,
- (ii) $x \leq y \Rightarrow \tau(x) \geq \tau(y)$,
- (iii) τ is continuous on $[0, s_{\max}]$.

Obviously, since a measure of similarity σ satisfies conditions (S1) and (S2) (cf. section 6.2.1.1), the application of τ to σ satisfies properties (D1*) and (D2). Thus, given a similarity function σ a dissimilarity function δ_σ derived by σ can be defined by

$$\delta_\sigma(d_a, d_b) =_{\text{def}} \tau(\sigma(d_a, d_b)). \quad (6-3)$$

A similarity-to-dissimilarity transformation τ is a parameter in the basic framework which allows to influence the 'degree of structuring' of a collection of documents. The remainder of this section presents a selection of possible transformation functions. In every case it is easy to check that the general conditions as defined above apply to the respective conversion function τ .

The most straightforward way is to choose a *linear conversion*

$$\tau_{\text{linear}}(x) =_{\text{def}} s_{\max} - x. \quad (6-4)$$

The advantage of a linear conversion is that the degree of similarity between given documents is reflected most faithfully by the resulting dissimilarity value. If a certain weighting of similarity values is desired there is a variety of functions to choose from. For example, a polynomial transformation

$$\tau_\lambda(x) =_{\text{def}} s_{\max} - x^\lambda, \lambda > 0, \quad (6-5)$$

can stress dissimilarity or similarity, respectively, to a degree defined by a parameter λ .

In order to stress the differentiation between groups of documents a cosine-based transformation can be defined by

$$\tau_{\text{cosine}}(x) =_{\text{def}} \frac{d_{\max}}{2} \cdot \left(1 + \cos \frac{\pi \cdot x}{s_{\max}} \right), \quad (6-6)$$

where d_{\max} denotes the maximal distance value desired for the transformation. Figure 6-3 shows the graphs of the respective similarity-to-dissimilarity transformations.

Similar to weighting similarity values in transformations, dissimilarity functions δ can be weighted by a suitable function $\omega : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$. Such a weighting function should have the following properties:

- (i) $\omega(0) = 0$,
- (ii) $x \leq y \Rightarrow \omega(x) \leq \omega(y)$,
- (iii) ω is continuous on \mathbb{R}_0^+ .

For example, $\omega_\lambda(x) =_{\text{def}} x^\lambda$, $\lambda > 0$, would be a suitable weighting function which stresses closeness or distance, respectively. Note that for metrics such a weighting may result in a violation of the triangular inequality (D3) since $x \leq y + z$ does not imply $x^\lambda \leq y^\lambda + z^\lambda$.

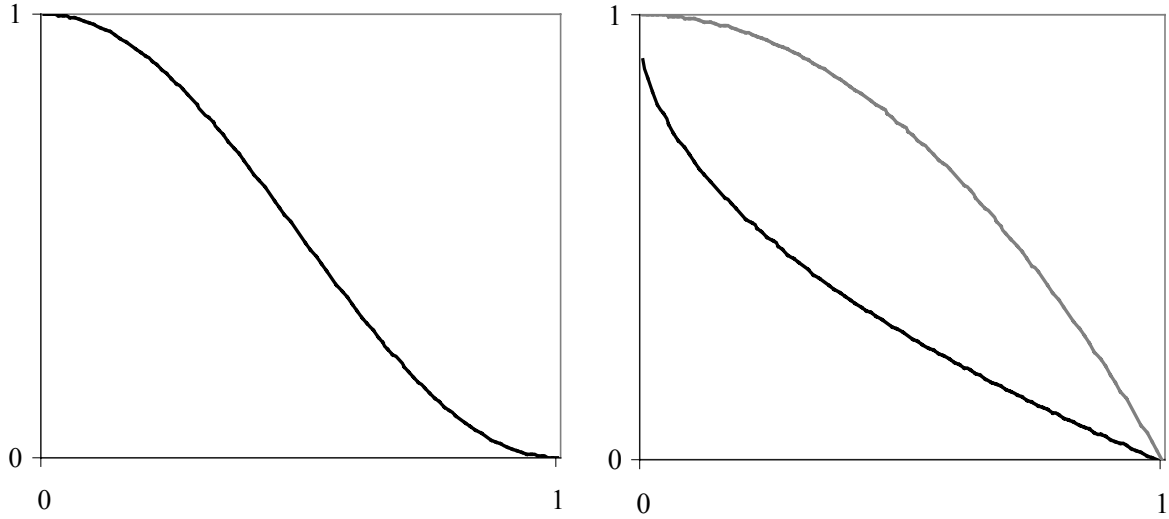


Figure 6-3: (a) Cosine-based transformation according to equation (6-6) with $s_{\max} = d_{\max} = 1$, (b) polynomial transformation according to (6-5) with $s_{\max} = 1$ and $\lambda = 0.5$ (black), $\lambda = 2$ (gray)

Of course, similarity values could be linearly transformed into dissimilarity values first and weighted afterwards. In any case, the result of this transformation or weighting step is a measure of dissimilarity $\delta: D \times D \rightarrow \mathbb{R}_0^+$ which yields high values for dissimilar documents and low values for similar documents. The function δ does not necessarily respect the triangular inequality.

6.2.2 The Spatial Scaling Module

The spatial scaling module (cf. figure 6-4) calculates a representative for each document in an m -dimensional space so that a reasonable amount of the dissimilarity information about the documents is preserved. More precisely, given n objects $d_1, \dots, d_n \in D$, a dissimilarity measure $\delta: D \times D \rightarrow \mathbb{R}_0^+$ for each pair of objects, and a target dimensionality m , the spatial scaling module constructs a set of vectors $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^m$ where each object $d_i \in D$ is represented by an $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$ with the property that the error s as defined by

$$s^2 =_{\text{def}} \sum_{i,j} (\delta^*(\mathbf{x}_i, \mathbf{x}_j) - \delta(d_i, d_j))^2 \quad (6-7)$$

is ‘fairly low’. In the formula δ^* denotes the distance measure used in the target space; s is called ‘stress’. In an optimal case, s would be minimized. In general, this is exactly the task of multidimensional scaling (as discussed in chapter 4.1).

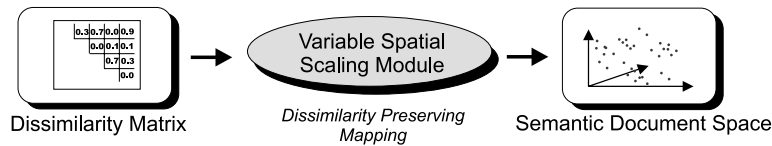


Figure 6-4: Context of the spatial scaling module within the basic framework

In this work the geometric scaling algorithm ‘FastMap’ developed by Faloutsos and Lin [FaLi95] is applied (cf. section 4.1.4). The reasons for choosing FastMap are as follows: First, the algorithm has attractive features when it comes to adding objects to the space (cf. sections 4.1.4 and 7.4.3). Furthermore, FastMap is a fast scaling method which is able to perform the mapping in linear time and does not require the calculation of the complete dissimilarity ma-

trix. Rather, document dissimilarities can be calculated on-the-fly when required by the algorithm – which is advantageous for complex document comparison techniques. Besides, the method is easy to implement (and thus yields a suitable basis for building a prototypical realization of the basic framework). In the case of FastMap, a ‘fairly low’ stress value means that s is not necessarily the minimal error value which is achievable but a reasonable approximation with respect to computational effort.

A disadvantage of the approach is that it theoretically requires the dissimilarity measure $\delta: D \times D \rightarrow \mathbb{R}_0^+$ to respect the properties of a metric (i.e. in particular the triangular inequality holds). A discussion of applying FastMap to non-metrical data is presented in section 6.2.2.2. Another drawback of the method is that it requires more dimensions than MDS for producing scaling solutions with comparable quality (for details see [FaLi95]). Since the constructed space will not be visualized directly this is no principal problem. However, the number of dimensions of the semantic document space effects the time necessary for training the self-organizing map in the final step (cf. section 6.4.6). Altogether, an optimal tuning of the basic framework with respect to the spatial scaling method used also depends on the realized document analysis module and the intended application. In the following, the FastMap algorithm and its application to non-metric data is presented.

6.2.2.1 The FastMap-Algorithm as an Example of a Spatial Scaling Module

The basic idea of FastMap is to interpret the objects $d_i \in D$ as points in a multi-dimensional space with an unknown dimensionality K . Then, the d_i are projected onto m mutually orthogonal axes (i.e. a Cartesian coordinate system) by only using the distance information provided by δ . FastMap uses the Euclidean distance as distance measure δ^* in the target space.

As a first step consider the problem for $m = 1$. The first problem to solve is to find an appropriate axis where the objects can be mapped to. The algorithm constructs an axis which goes through two selected objects d_a and d_b , called ‘pivot objects’, in the original (virtual) multi-dimensional space. The question of how to choose the pivot objects will be discussed later. Figure 6-5 illustrates the idea of the mapping: Given a line through the pivot objects d_a and d_b each object $d_i \in D$ has to be ‘projected’ to this axis. The (fixed) objects d_a, d_b and each single d_i define a triangle where the length of each side is given by the distance function δ . By an orthogonal projection to the line the new distance of the ‘image’ of d_i to d_a and d_b , respectively, preserves some information about the distance of these objects in the original space. Denote the distance of d_i ’s image to d_a as x_{ij} (the meaning of the index j will become clear later). Obviously, if $\delta(d_a, d_i)$ is rather small, x_{ij} will be small, too. Thus, in many cases the proportion of x_{ij} to $(\delta(d_a, d_b) - x_{ij})$ will roughly reflect the proportion of $\delta(d_a, d_i)$ to $\delta(d_b, d_i)$, though more or less stress will be introduced.

The desired orthogonal projection of each d_i to the line can be done by applying the Pythagorean theorem to both right-angled triangles (cosine law) and solving the resulting equation by x_{ij} which yields

$$x_{ij} = \frac{\delta^2(d_a, d_i) + \delta^2(d_a, d_b) - \delta^2(d_b, d_i)}{2 \cdot \delta(d_a, d_b)}. \quad (6-8)$$

Since the x_{ij} ‘store’ some information about the distance between object d_i and the current pivot objects, each x_{ij} can be saved as the j -th component of \mathbf{x}_i (which is the representative of d_i in \mathbb{R}^m).

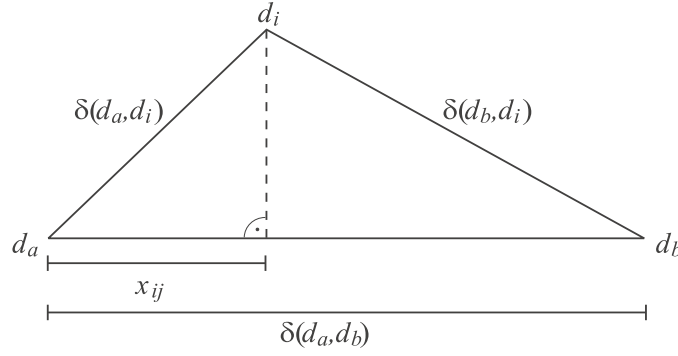


Figure 6-5: Projecting objects to a single axis (illustration adapted from [FaLi95])

Obviously, $x_{aj} = 0$ and $x_{bj} = \delta(d_a, d_b)$. It is crucial that the distance of the pivot objects is maximal with respect to all distances between objects. If e.g. $\delta(d_a, d_i) < \delta(d_a, d_b) < \delta(d_b, d_i)$, the result of equation (6-8) might yield an $x_{ij} < 0$ which would produce a heavy distortion of the original distance information. Searching for the maximal distance would require $O(n^2)$ distance computations. However, since the authors of FastMap strive for a linear-time algorithm, this step is prohibitive. Instead, the authors propose a linear-time heuristic algorithm to search for distant objects (algorithm 6-1). The actual search is repeated a constant number of times to improve the likelihood that the selected objects are indeed suitable.

Having done the mapping for one axis, an extension of this method for an arbitrary m requires some additional effort. Following the basic idea of the approach, each object d_i is considered as a point in some K -dimensional space. An additional axis which is orthogonal to the axis through the selected pivot objects d_a and d_b can be found in a hyper-plane H which is perpendicular to the line through d_a and d_b . In this hyper-plane the same mapping as described above can be applied. Prior to repeating the mapping, however, the objects d_i have to be projected onto the $(K-1)$ -dimensional hyper-plane. More precisely, the distance function δ has to be ‘embedded’ into this subspace. Figure 6-6 depicts this situation. Applying the Pythagorean theorem to the triangle $d_i C d_k$ yields the desired embedding δ' :

$$(\delta'(d_i, d_k))^2 = (\delta(d_i, d_k))^2 - (x_{ij} - x_{kj})^2 \quad (6-9)$$

for $1 \leq i, k \leq n$. These two steps – projection to an axis and embedding of δ – can be iterated m times, thus yielding a Cartesian space where $\delta^*(x_i, x_k)$ approximates $\delta(d_i, d_k)$. For the embedding in step j , $1 \leq j \leq m$, the current distance information can be recursively computed as

$$\begin{aligned} \delta_1(d_i, d_k) &= \delta(d_i, d_k) \\ (\delta_j(d_i, d_k))^2 &= (\delta_{j-1}(d_i, d_k))^2 - (x_{ij-1} - x_{kj-1})^2 \end{aligned} \quad (6-10)$$

Algorithm 6-1: Searching for distant objects [FaLi95]

```

choosePivots ( $\delta(\cdot, \cdot), D$ )
BEGIN
     $d_b :=$  arbitrary  $d_i \in D$ ;
    FOR  $i := 1..const$  DO BEGIN                // e.g. const = 5
         $d_a := d_b$ ;
        FOR  $k := 1..n$  DO                      // for all objects
            IF  $\delta(d_a, d_k) > \delta(d_a, d_b)$  THEN  $d_b := d_k$ ;
        END;
    return  $d_a, d_b$ ;
END

```


value of the greatest violation of the triangular inequality to each dissimilarity value of δ (cf. [Davi83]). More precisely, given a dissimilarity measure $\delta: D \times D \rightarrow \mathbb{R}_0^+$ for which properties (D1) and (D2) from section 6.2.1.1 hold but the triangular inequality (D3) is violated, perform the following steps for transforming δ into a metric δ' :

- Compute $e =_{\text{def}} \max \{ \delta(d_a, d_b) - \delta(d_a, d_c) - \delta(d_c, d_b) \mid d_a, d_b, d_c \in D \}$. Note that $e > 0$ if the triangular inequality is violated.
- Compute δ' as $\delta'(d_a, d_b) =_{\text{def}} 0$ if $d_a = d_b$ and $\delta'(d_a, d_b) =_{\text{def}} \delta(d_a, d_b) + e$ otherwise.

Of course this procedure adds some stress to the originally given dissimilarity information which is inevitable anyway, since a *metric* space will be constructed to ‘reflect’ δ . But more substantially, the transformation of δ into a metric δ' as presented above would require a computational effort of $O(|D|^3)$ which is prohibitive for some of the document collections interesting in this work. The following heuristic tries to determine a good value for e in time $O(|D|^2)$:

- Choose a random object d .
- Compute $e =_{\text{def}} \max \{ \delta(d, d_a) - \delta(d, d_b) - \delta(d_a, d_b) \mid d_a, d_b \in D \}$.

Repeating this procedure a constant number of times and choosing the greatest error value e improves the quality of the results. Experiments with 6 data sets (containing 420 documents in average) have shown that the error value computed by this heuristic is not dramatically lower than the true error: For 10 (30) iterations the method achieves an error value which is about 30% (20%) lower than the true error. In other words, the heuristic does not ‘close’ the distance gap between all objects which violate the triangular inequality but lowers it at least significantly, thus yielding a more solid basis for applying FastMap. However, given the higher complexity which comes along with this step, other multidimensional scaling methods could be used instead as well.

Besides adjusting the input data in a computationally expensive way, is there a suitable adaptation of FastMap which deals with non-metrical dissimilarity data more correctly? A deeper discussion of that would be beyond the scope of this work. However, an *ad-hoc* approach can be realized as follows:

Obviously, there are two crucial steps in the algorithm which have to be considered: projecting objects to a single axis and embedding δ in the hyper-plane. First, equation (6-8) applies the cosine law for projecting an object to the pivot axis. If the current pivot objects d_a, d_b and object d_i , which is to be mapped to the axis, violate the triangular inequality the computed value x_{ij} is not optimally defined. In such a case there is a gap e between the respective distances, i.e. $e =_{\text{def}} \delta(d_a, d_b) - \delta(d_a, d_i) - \delta(d_b, d_i) > 0$. Instead of applying equation (6-8) in such a case, the locally optimal solution is to define $x_{ij} =_{\text{def}} \delta(d_a, d_i) + e/2$. Implicitly, by doing this the current distances $\delta(d_a, d_i)$ and $\delta(d_b, d_i)$ are increased by $e/2$. This change has to be considered when embedding δ in the hyper-plane for the next iteration according to equation (6-10). To realize this adaptation efficiently a sparse $n \times m$ -matrix A can be used where $A(i, k)$ stores the error value $e/2$ for the distance of an object d_i to the (fixed) pivot objects in step k .

Experiments have shown that the stress produced by the modified FastMap algorithm for non-metrical dissimilarity information does not significantly differ from the stress produced by the original algorithm. More details on applying the original algorithm to metrical and non-metrical input data can be found in section 6.4.1.

6.2.3 Topology Preserving Mapping and Visualization Module

The final module of the basic framework (cf. figure 6-7) is realized by a self-organizing feature map and a suitable visualization method. A self-organizing feature map (SOM), a neural network model developed by Teuvo Kohonen [Koho82, Koho95], projects an m -dimensional input space into a d -dimensional sub-space. More precisely, a SOM realizes a mapping of nonlinear statistical relationships between high-dimensional input vectors to a grid structure of network units with the property that the distance relationships between the input patterns are preserved as good as possible. In most applications a 2-dimensional rectangular grid is used, but also different grid layouts and output dimensions are possible, e.g. a hexagonal 2D grid or a 3D torus. Note that the general model allows different implementations and adjustments. Many proposals for parameter settings and realizations of the basic model can be found in literature (e.g. [Zell94, Brau95]). Ultimately, a concrete implementation depends on the application and needs some individual adjustments. The realization of the SOM in this work – which is described in this section – is guided by the implementation of S. Sklorz [Sklo96, Sklo01] (with only slight modifications).

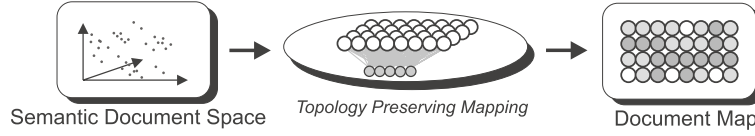


Figure 6-7: Context of the topology preserving mapping module within the basic framework

6.2.3.1 Architecture and Training of a Self-Organizing Feature Map in this Work

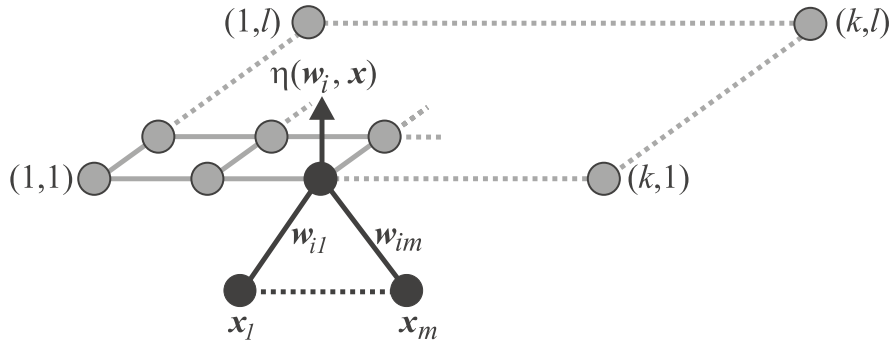
A SOM consists of one layer of active units which are disposed in a d -dimensional grid structure. More specifically, in this work a two-dimensional $k \times l$ -rectangular grid is used where each unit $i = (i_1, i_2)$, $1 \leq i_1 \leq k$, $1 \leq i_2 \leq l$, has at most 4 direct neighbors $N(i)$ defined by

$$N(i) =_{\text{def}} \left\{ (i'_1, i'_2) \left| \begin{array}{l} (i'_1 = i_1 \pm 1, i'_2 = i_2, 1 \leq i'_1 \leq k) \vee \\ (i'_1 = i_1, i'_2 = i_2 \pm 1, 1 \leq i'_2 \leq l) \end{array} \right. \right\}. \quad (6-11)$$

The units in the grid are not connected with each other. However, the local neighborhood relation $N(i)$ defines a spatial arrangement of the units. Each unit i in the grid is linked with all m units of the input layer by means of m weighted edges, formally realized by a weight vector $\mathbf{w}_i = (w_{i1}, \dots, w_{im}) \in \mathbb{R}^m$. This weight vector represents the current state of the respective unit. For a final set of input patterns $X \subset \mathbb{R}^m$, each $\mathbf{x} \in X$ is handed over directly and in parallel to all units i of the grid. Thus, at time t all units i receive the same stimulation \mathbf{x} . In response to this stimulation, each unit i calculates the output function $\eta(\mathbf{x}, \mathbf{w}_i)$ which measures the similarity or distance between the input pattern \mathbf{x} and its own state, described by the weight vector \mathbf{w}_i . In this work the Euclidean distance is used, i.e.

$$\eta(\mathbf{x}, \mathbf{w}_i) =_{\text{def}} \sqrt{\sum_{h=1}^m (x_h - w_{ih})^2}. \quad (6-12)$$

For the purpose of this work it is crucial that the number of units in the grid exceeds the number of input patterns by an order of magnitude (details on that will be discussed in section 6.4.2). Figure 6-8 depicts the network's architecture chosen in this work. The solid gray lines between units symbolize the neighborhood relation.

Figure 6-8: Architecture of a Kohonen map with a $k \times l$ grid

During its unsupervised training the SOM tries to ‘capture’ the topological structure of the input space by means of adjusting the states \mathbf{w}_i of all units i in such a way that (a) for every input pattern \mathbf{x} there is a corresponding unit i where \mathbf{w}_i is very similar to \mathbf{x} , i.e. i represents \mathbf{x} , and (b) neighboring units (with respect to a notion of distance on the grid based on $N(i)$) represent neighboring input patterns (with respect to the distance of the objects in the input space, in this case the Euclidean distance).

At the beginning of the training phase random values are assigned to the units’ weight vectors. During the learning process the input patterns \mathbf{x} are presented several times and in random order to the network for a certain period of time which is defined by a fixed number of discrete learning steps t_{\max} .

In detail, the learning process includes the following steps: At each time t , $0 \leq t \leq t_{\max}$, $t \in \mathbb{N}$, a single unit i^* is determined for the current input vector \mathbf{x} where the unit’s weight vector \mathbf{w}_i is most similar to \mathbf{x} , i.e.

$$i^* = i \Leftrightarrow_{\text{def}} \|\mathbf{x} - \mathbf{w}_i\| = \min\{\|\mathbf{x} - \mathbf{w}_j\|, \forall \text{ units } j\}. \quad (6-13)$$

Such a best matching unit i^* is called the *cluster center* of the considered object vector. The weight vector \mathbf{w}_{i^*} of the cluster center and the weight vectors of units in a certain topological surrounding now are shifted towards the input vector. The amount of shifting depends on a learning rate, the distance between \mathbf{w}_i and \mathbf{x} , and the unit’s position in the area surrounding the cluster center. Both, learning rate and size of the neighborhood area decrease in time.

Formally, let $\mathbf{w}_i(t)$ denote the weight vector of unit i at time t . Then, for all units i in the grid the respective weight vector $\mathbf{w}_i(t+1)$ for the next step is defined as

$$\mathbf{w}_i(t+1) =_{\text{def}} \mathbf{w}_i(t) + \alpha(t) \cdot v(i, i^*, t) \cdot (\mathbf{x} - \mathbf{w}_i(t)). \quad (6-14)$$

$v(i, i^*, t)$, $0 < \delta(\cdot, \cdot) \leq 1$, is called *neighborhood kernel* and determines the degree to which a given unit i of the grid belongs to the neighborhood of the cluster center i^* at time t . Here, the neighborhood kernel has the shape of the Gaussian function $f(x) = 2\pi^{-1/2} \cdot \exp(-x^2/(2\sigma^2))$, where the standard deviation σ is replaced by a time-dependent, monotonously decreasing radius function $r(t)$ which converges against zero. As a result the neighborhood kernel’s width is successively narrowed. Formally, v is defined as

$$v(i, i^*, t) =_{\text{def}} \exp\left(-\frac{\|i - i^*\|^2}{2r^2(t)}\right), \quad (6-15)$$

where the norm $\|\cdot\|$ measures the distance of units i and j on the grid, i.e.

$$\|i - j\| =_{\text{def}} \max\{|i_1 - j_1|, |i_2 - j_2|\}. \quad (6-16)$$

The radius function $r(t)$ is defined as

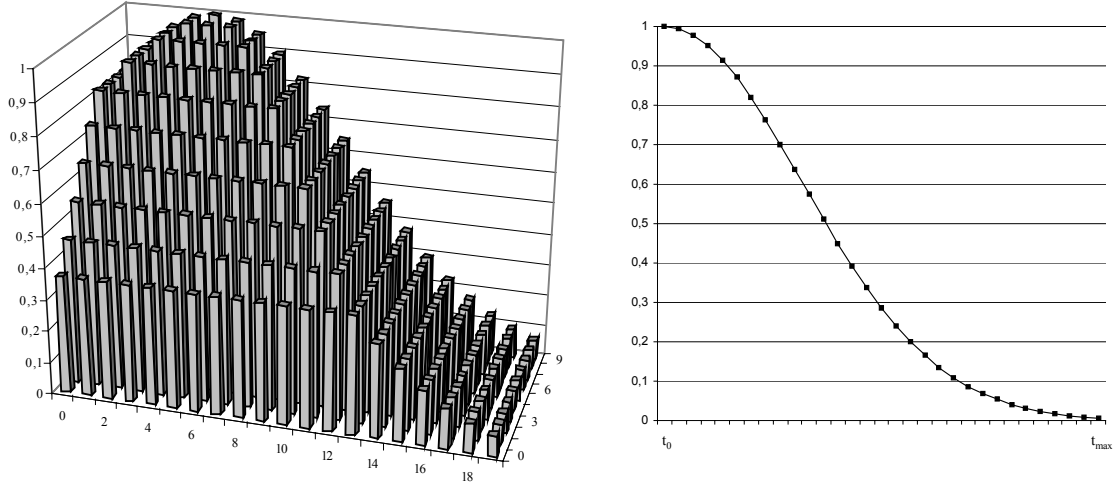


Figure 6-9: (a) Neighborhood kernel $v(i, i^*, t)$ on a 10×20 grid with cluster center $i^* = (4, 9)$ at time $t = 15$, $t_{\max} = 30$
 (b) graph of learning rate $\alpha(t)$ with $t_{\max} = 30$ and $\gamma = 5$

$$r(t) =_{\text{def}} r_{\max} \cdot \exp\left(-\frac{\gamma t^2}{t_{\max}^2}\right), \quad (6-17)$$

where γ is a suitable constant, in this case $\gamma = 5$, that defines how steep r will descend.

Note that $v(i, i^*, t) = 1 \Leftrightarrow i = i^*$, i.e. the cluster center i^* is the center of the neighborhood kernel and thus receives maximal adjustment. The initial size of the neighborhood kernel – i.e. the initial value of $r(t)$ in v – is crucial: In the beginning an extensive neighborhood kernel is necessary so that even units in a large distance can affect each other. Otherwise the map may suffer from topological defects, i.e. parts of the map are ordered correctly whereas the global structure is faulty (cf. [Zell94] for an example). To ensure that the complete map is affected by the initial learning steps r_{\max} is defined with respect to the chosen grid layout as

$$r_{\max} =_{\text{def}} \sqrt{k^2 + l^2}. \quad (6-18)$$

The *learning rate* $\alpha(t)$, $0 < \alpha(\cdot) \leq 1$, is a function that controls the strength of weight adjustments during learning. In the initial learning steps large corrections are desired to capture the course structure of the input space whereas later only a moderate refinement of the units' weight vectors is reasonable so that the network gradually reaches a stable state. It is recommendable to synchronize the function types of α and v in order to avoid undesired mutual 'interference'. Thus, α is defined as

$$\alpha(t) =_{\text{def}} \exp\left(-\frac{\gamma t^2}{t_{\max}^2}\right). \quad (6-19)$$

Learning rate and neighborhood kernel, in combination, are a means to control the degree to which units in the grid are influenced by the learning process. The normalization of function values to the interval $]0;1]$ guarantees that the network state converges, i.e. distances between weight vectors of the grid units and input patterns continuously decrease. Figure 6-9 visualizes both functions with some fixed parameters. Figure 6-10 depicts the adjustment of two-dimensional weight vectors in a simple 1×7 grid. Note that the degree to which vectors are shifted depends only on the neighborhood relation between each unit i and the cluster center i^* as well as the current learning rate, but not on the degree of similarity between i and i^* . The learning process of the SOM is summarized in algorithm 6-3.

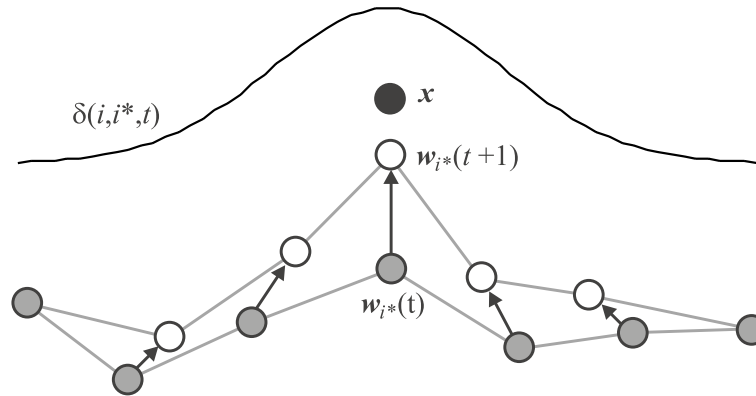


Figure 6-10: Illustration of weight vector adjustment during the learning process at time t . Grey lines symbolize the neighborhood relation of the weight vectors

To briefly sum up the effect of the learning process: Weight vectors w_{i^*} of best matching units i^* for a certain input pattern x will get even more similar to x during training. With a decreasing strength this holds for units in the surrounding of i^* , too. Consequently, after the learning process it can be expected that (a) similar objects from the input space X will have corresponding cluster centers which are located ‘closely’ to each other in the grid, i.e. the relative positions of the cluster centers correspond to the similarity structure of the input patterns; and (b) units i belonging to similar weight vectors w_i will have similar positions in the grid, i.e. the weight vectors are ordered according to their similarity.

As stated above the number of units in the grid exceeds the number of objects from the input space (details on that can be found in section 6.4.2). Consequently, the set of grid units can be partitioned in a set of cluster centers and a set of units which do not appear as cluster centers. In other words, for many units there are no corresponding input patterns in X . These units can be seen as generalized input patterns which represent a priori unknown inputs. As the next section shows, they provide additional information about the global structure of the input space.

6.2.3.2 Visualizing the SOM

In order to visualize the structure of the input space X which has been mapped to the two-dimensional grid it is necessary to ‘extract’ the topological information encoded in the SOM. In this work the P -matrix visualization method introduced by Sklorz [Sklo96, SBJ99] is used (cf. chapter 4.2.3). This method directly uses the topological properties of the trained neural network as well as the information gained by generalization. The reason for selecting the P -matrix visualization method is mainly its intuitive presentation of fine-granular neighborhood

Algorithm 6-3: Training of a SOM

```

initialize ARRAY  $W[1..k, 1..l]$  OF WEIGHT VECTORS
with small random values;
FOR  $t := 0..t_{\max}$  DO BEGIN
    randomly choose an input pattern  $x$ ;
    compute cluster center  $i^*$  for  $x$  according to equation (6-13);
    FOR all units  $i$  in the grid DO
        adjust weight vector  $W[i_1, i_2]$  according to equation (6-14);
END;
```

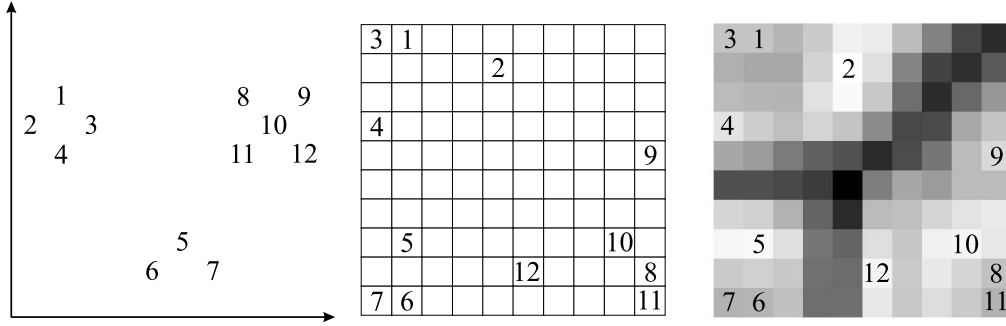



Figure 6-11: (a) Arrangement of 12 objects in a 2D input space, (b) SOM-computed mapping (240 learning cycles) to a 10×10 grid, (c) visualization of P -matrix of the SOM

relationships between objects in high-dimensional spaces: In contrast to other distance-matrix methods (cf. chapter 4.2.3) each visualized grid unit has a direct meaning with respect to the input patterns. Moreover, the visualization method yields good results even for rather weakly trained SOMs, thus saving computation time. Finally, the approach has been successfully used for interpreting the distribution of single data objects for data mining purposes (cf. [SBJ99, Sklo01]), suggesting its high potential for the application domain of this work.

The P -matrix is a mapping $P: \{1, \dots, k\} \times \{1, \dots, l\} \rightarrow \mathbb{R}_0^+$ which yields for every unit $i = (i_1, i_2)$ the value of the greatest similarity between its weight vector \mathbf{w}_i and each input pattern \mathbf{x} , formally

$$P(i_1, i_2) =_{\text{def}} \min \{ \eta(\mathbf{x}, \mathbf{w}_i) \mid \mathbf{x} \in X \}. \quad (6-20)$$

In other words, P superimposes all stimulation patterns which are produced by presenting the single vectors $\mathbf{x} \in X$ to the neural network. Therefore, P reflects the density of the input space X which is encoded in the weight vectors \mathbf{w}_i of the trained SOM: Units in the grid with weight vectors which are rather dissimilar to all vectors from the input space were trained rarely according to the learning algorithm. Those weight vectors represent the ‘empty space’ between clusters of input patterns. In contrast, clusters of similar input patterns lead to a more intensive training of a particular region in the grid. The weight vectors in these regions can be expected to be close to each other and their minimal distance to the input patterns will be rather low. Thus, high values in P separate regions of similar objects. A deeper theoretical discussion regarding the density of weight vectors and input patterns can be found in [Ritt91, Koho95].

The information provided by P can be visualized by assigning a shade of gray to the values in P : Using a linear function that maps the minimal value to white and the maximal value to black it is possible to visually detect regions of similar input patterns. These regions will be displayed as bright shaded areas, separated by dark shades corresponding to large values in P . Though the gray shades alone provide a good deal of information about the global structure of the input space it is useful to mark those units in the grid which correspond to the cluster centers i^* of the input patterns \mathbf{x} according to equation (6-13). Figure 6-11 presents a mapping of objects from a simple input space to a SOM grid and the visualization of the corresponding matrix P .

6.2.4 Summary of the Basic Map Generation Process

The process diagram in figure 6-12 summarizes the computation of a document map, taking into account the exchangeability of the document analysis and the spatial scaling module. The mathematical interfaces of the variable modules have been presented in sections 6.2.1 and 6.2.2. Each possible document analysis module indexes the given collection of documents and

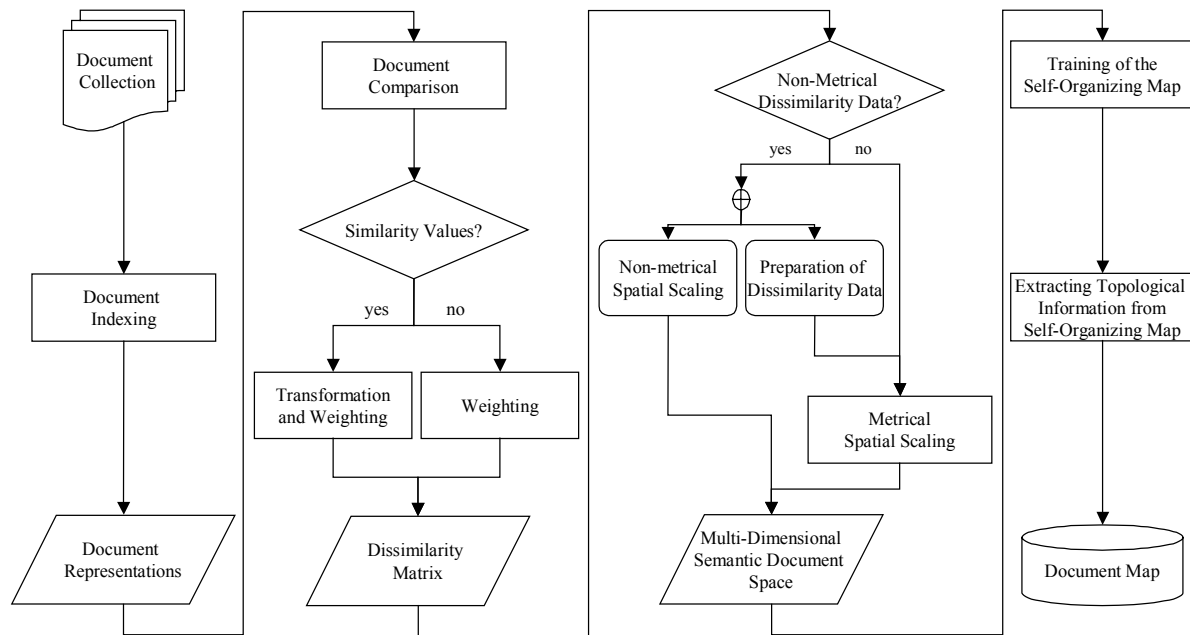


Figure 6-12: Process diagram of the basic document map generation

computes a set of formal document representations; the actual document comparison is performed on this data structure. The document matching function may yield similarity or dissimilarity values. Similarity values are transformed into dissimilarity values, along with a weighting if desired (cf. section 6.2.1.5). Similarly, dissimilarity values may be weighted (if no weighting is desired the weighting process simply realizes the identity function). The resulting dissimilarity matrix may contain metrical as well as non-metrical dissimilarity data. For setting up the semantic document space in the first case metrical spatial scaling methods (cf. section 6.2.2.1) can be applied. In the latter case either non-metrical scaling techniques are used or the given dissimilarity data is pre-processed (cf. section 6.2.2.2). The resulting multidimensional semantic document space provides the training data for the self-organizing map (cf. section 6.2.3.1). Finally, the topological information contained in the neural network is extracted and visualized as described in section 6.2.3.2.

6.3 Interpreting a Document Map

Having presented the basic framework for generating document maps, this section discusses the metaphor of the resulting graphical presentation in more detail. In a document map (see figure 6-13 for an example³) documents are represented as points. The map conveys a picture of the cluster structure of the semantic document space, i.e. the grouping of similar documents and distribution of related document groups. The concept of ‘document similarity’ – according to an appropriate measure – is reflected by a notion of neighborhood: Neighborhooded points in common bright shaded areas represent similar documents. These areas are separated by gray borders: the darker the border, the stronger the separation, and thus the more dissimilar the single documents or document groups. More intuitively, the semantics of this map can be described by the metaphor of ‘mountains’ and ‘canyons’: One can imagine that the bright

³ The figure shows a document map consisting of 137 articles from the online version of the ‘Süddeutsche’ newspaper (www.sueddeutsche.de). Articles from the category ‘university’ can be found in the northern half, articles related to ‘traveling’ are located in the southern half.

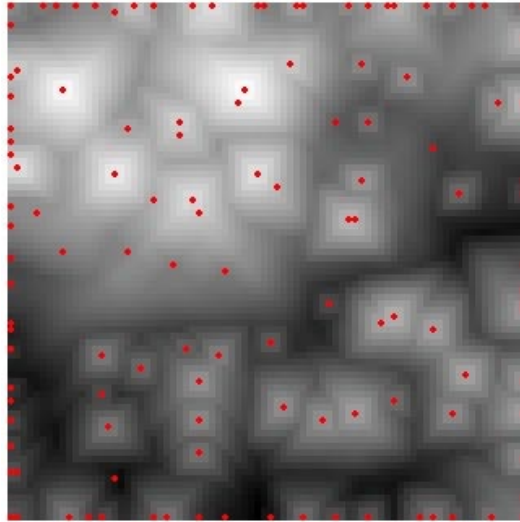


Figure 6-13: A sample document map. The points represent documents, related documents are grouped in bright-shaded areas, separated by borders with different strength which corresponds to the degree of dissimilarity.

shaded areas are ‘mountain plateaus’ on which similar documents are located, possibly further grouped by ‘rifts’ (bright shades of gray). The dark borders are ‘canyons’ which separate mountain plateaus. A dark border corresponds to a deep canyon and thus to a relatively large distance between two mountains. The closer two document points or groups appear in the map (taking into account the distance information provided by the shades of gray) the more similar the respective documents or text groups are.

Considering the properties of the mapping method (cf. chapter 4.2.2) some warnings are appropriate here: First, it is not possible to ‘measure’ the absolute degree of similarity of a pair of documents (or distance in the semantic document space, respectively) by considering the distance of the corresponding document points in the map display (not even in ‘white’ areas). This is due to the SOM’s property of globally optimizing neighborhood relationships rather than absolute distances. However, for analyzing the similarity structure of a text corpus knowledge about an absolute degree of similarity (which is an artificial number, anyway) is not necessary anyhow (cf. section 5.8.1). Second, though documents or groups of documents that are neighbored in the map display are also neighbored in the semantic document space (and thus similar according to the measure used), the converse does not hold: There may be neighbored (similar) clusters of documents in the semantic document space which are located in some distance in the map display. Such a stretching of object arrangements, however, is inevitable in any projection of inherently high-dimensional relationships (like similarity relationships of texts) to a low dimensional output space. Detecting relationships between document groups in such a situation remains an intellectual effort. An additional interpretation aid will be discussed in chapter 11.3.2.

A final remark concerns the apparently many document points which are located at the edges of the sample map in figure 6-13 – an effect which can be observed frequently when looking at the maps presented in the remainder of this work. Though the effect may be visually not appealing it produces no problems: The edges are interpreted in the same way as the rest of the map. Groupings of documents can still be detected in these areas without problems. Absolute distances on the output grid do not matter, anyway, so that the closeness of document points does not distort the overall information provided by the map. The effect itself is caused by the chosen rectangular grid layout (cf. section 6.2.3.1): Units that are located near the edges have an assymetric and relatively small neighborhood. Thus, there are less possibilities to arrange input patterns near the edges. Taking a sphere-layout instead

would avoid this effect – but then either a three-dimensional output device would have to be used or the sphere would have to be projected onto two dimensions again. In the latter case the interpretation of the map would become more difficult.

6.4 Setting Parameters for Generating Document Maps

The basic framework and its realization obviously have many parameters which have an effect on the quality of the mapping and the displayed structures (cf. table 6-1). It is thus important to summarize these parameters and to discuss the relevant effects they have on the resulting document maps. This section leaves the formal presentation of the framework and its realization and mainly discusses experimental results in order to make clear the effects of the different parameters. In the following the parameters of the framework's 'core', namely the spatial scaling and the topology preserving mapping module, will be discussed first.

As the spatial scaling module is variable in the proposed framework different scaling modules may have different adjustable parameters. The most important parameter in the framework's context, however, is the number of dimensions of the semantic document space. This number has direct influence on the quality of the semantic document space: The more dimensions are available, the more faithful the dissimilarity information – which is calculated by the document analysis module – can be preserved. However, the number of dimensions will also influence the computation time for generating the document map. Section 6.4.1 discusses this parameter for the scaling module used throughout this work. In section 6.4.2 hints for setting the parameters for training and visualization of the self-organizing feature map will be given. Since this module is fixed the SOM parameters are general framework parameters.

Another parameter is the transformation or weighting function for similarity or dissimilarity values, respectively. Details on that can be found in section 6.4.3. It will turn out that this parameter has direct influence on the 'degree' to which a semantic document space will be structured. This leads to a brief excursus in section 6.4.4 where a measure for the 'degree of structuring' will be developed. The document analysis module is also variable in the framework and the parameters to be influenced may vary from realization to realization. Since the vector space model as presented in section 6.2.1.3 is the 'default' realization and will be used for some features of the prototype system (cf. chapter 7) some properties of the respective parameter settings will have to be briefly discussed in section 6.4.5. This section concludes with a discussion on the complexity of the overall approach (6.4.6).

Table 6-1: Important adjustable parameters of the basic framework

module / interface	important adjustable parameters
document analysis module	<i>realization dependent</i>
weighting / transformation of dissimilarity / similarity values	choice of weighting / transformation function
spatial scaling module	number of dimensions of semantic document space ($ X = m$) + <i>realization dependent parameters</i>
self-organizing feature map	size of the grid ($k \times l$) number of training steps (t_{\max})

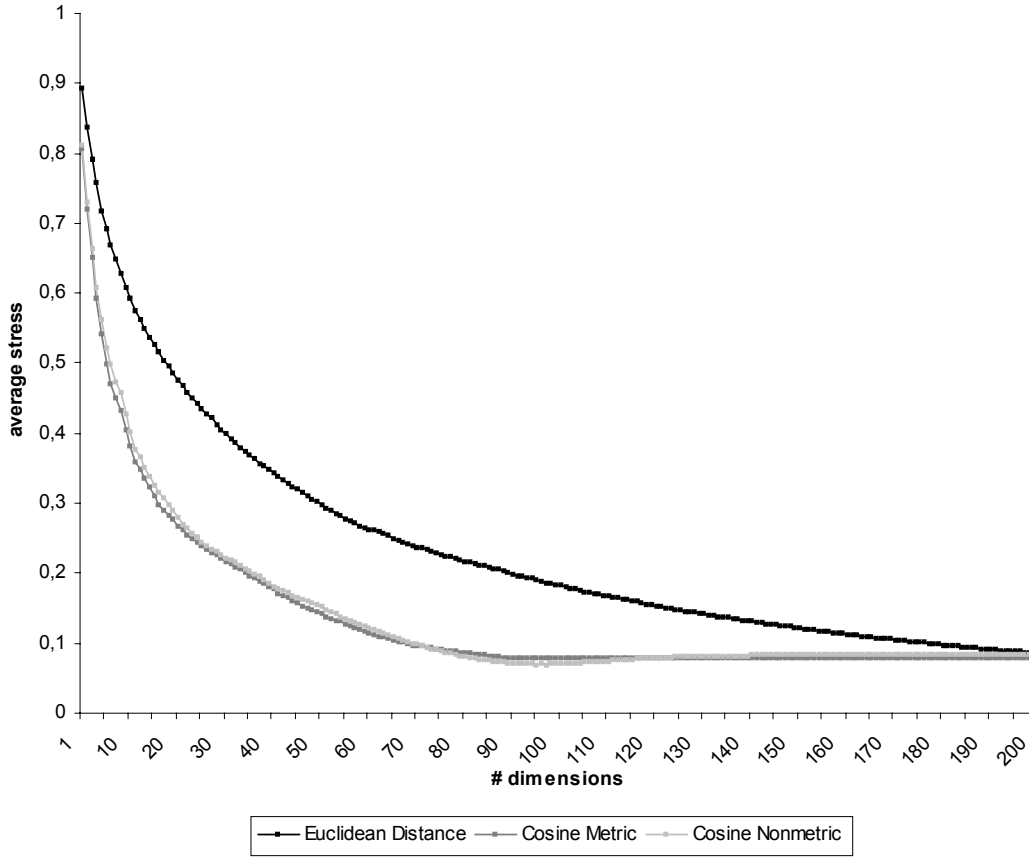


Figure 6-14: Stress curves for different (dis-)similarity measures

6.4.1 Performance of the Spatial Scaling Module

Crucial for the quality of the document map to be computed is the faithfulness with which the dissimilarity between each pair of documents as computed by the document analysis module is reflected in the semantic document space. The averaged error between the original dissimilarity data and the distances in the semantic document space is measured by the stress value (equation (6-7) in section 6.2.2). In the following the stress value is normalized for a convenient interpretation, i.e.

$$s^2 =_{\text{def}} \frac{\sum_{i,j} (\delta^*(x_i, x_j) - \delta(d_i, d_j))^2}{\sum_{i,j} (\delta(d_i, d_j))^2} \quad (6-21)$$

Quite clearly, the greater the number of available dimensions the better can the dissimilarity values be reflected in the target space. How many dimensions are necessary for a satisfying mapping? This section presents some experiments which help to get a better impression of how the spatial scaling performs.

For a given document collection and a fixed dissimilarity measure a stress curve $s: \mathbb{N} \rightarrow \mathbb{R}_0^+$ indicates the stress value $s(m)$ for a corresponding m -dimensional space. In fact, plotting the stress value against different numbers of dimensions is a common way to decide how many

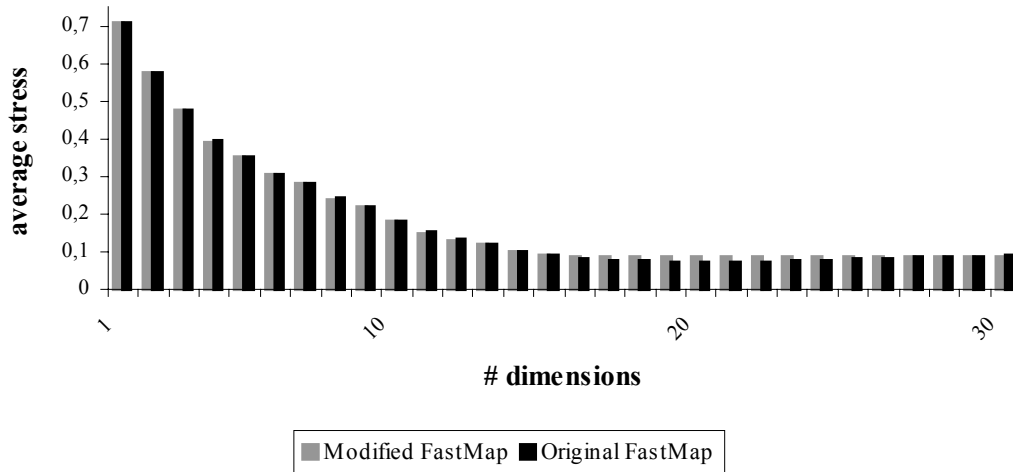


Figure 6-15: Stress values for the modified and original spatial scaling algorithm

dimensions to specify for an MDS problem [Tulsa97]. Figure 6-14 shows three stress curves for spaces that have been constructed according to the spatial scaling algorithm presented in section 6.2.2, each calculated for a particular dissimilarity measure. The curves have been averaged over four document collections which comprise 460 documents in average. The document analysis module applied is based on the vector space model as presented in section 6.2.1.3. In the vector space model documents are internally encoded as numerical term vectors. In this case the application of the spatial scaling module serves as a dimension reduction technique: In average, the term vector space is spread up by approximately 1,700 indexing terms (or dimensions). The term vectors contain approximately 40 non-zero entries in average. Though the number of dimensions may be reduced by an additional trimming of the term vectors so that terms that appear only in one document are removed, in practical applications there are still too many dimensions for applying the topology preserving mapping module.

The document matching functions used in the presented experiments are the Euclidean distance and the cosine measure of similarity. The latter has been converted to a measure of dissimilarity by a linear transformation (cf. equation (6-4) in section 6.2.1.5). Note that the resulting cosine-based measure of dissimilarity does not satisfy the triangular inequality and, thus, is no metric whereas the Euclidean distance satisfies the properties of a metric. There are two stress curves for the cosine measure. For the first the dissimilarity information has been used directly as input to the spatial scaling module, for the second it was converted into metric distance data (using the exact error e , cf. section 6.2.2.2).

The stress curve for the Euclidean distance is strictly monotonously decreasing as expected. In fact, experiments show that the spatial scaling module achieves nearly zero stress if the target space has enough dimensions. For the metrical cosine-based dissimilarity measure the averaged minimal stress which could be achieved is approximately 7%. The curve does not further decrease after 97 dimensions. It is obvious that zero stress is not achievable since non-metrical dissimilarity data has been embedded in a metrical space. Quite surprisingly, for the non-metrical data the spatial scaling method – though designed for metrical distance data – performs well, too. However, at some point the corresponding stress curve slightly increases. This is due to the fact that in this case the embedding of δ as realized by equation (6-10) from section 6.2.2.1 causes an additional error which is cumulative over time. More precisely, applying the equation may yield negative results for $(\delta_j(d_i, d_k))^2$ which are set to zero in the realization. The increasing of stress stops once all considered distances are zero (in all experi-

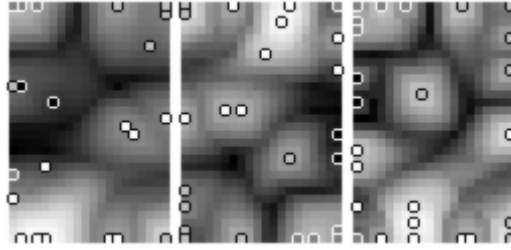


Figure 6-16: Visualizations of a self-organizing map trained with (a) 10, (b) 20, and (c) 40 learning steps per input pattern

ments the curve does not increase dramatically). These stress curves provide evidence that the spatial scaling method applied here yields reasonable and useful results for both, metrical and non-metrical data. The adaptation of the scaling method for non-metrical data, though theoretically more satisfying, does not produce a relevant improvement. For a collection of 168 documents and the non-metrical, linearly transformed cosine measure of similarity figure 6-15 compares the stress values achieved with the modified and original version, respectively, of the spatial scaling algorithm (cf. section 6.2.2.2).

6.4.2 Setting Parameters for the Topology Preserving Mapping

In order to exploit the advantages of the visualization technique applied in this work it is necessary to choose an adequate size of the neural network's grid. Recall that neurons in the grid that do not correspond with an input pattern x from the semantic document space X are generalized patterns. Furthermore, according to the learning algorithm, well-trained map areas represent densely 'populated' areas of the input space. These areas will be shown as bright shaded sectors in the visualized map whereas poorly trained, dark sectors represent empty or sparsely populated space between clusters of input patterns. Provided that there are enough neurons to represent generalized patterns the visualization helps to easily understand the structure of the space. Experiences with this approach show that the number of input patterns should ideally not exceed 20% of the number of neurons in the grid – otherwise it becomes hard or even impossible to identify relationships on a document-to-document level. Though there is no constraint for the upper bound of the net size, a rule-of-thumb is that using less than 5% of the neurons in the grid as representatives for input pattern is a waste of computation time since no additional structural information or ease of interpretation can be gained.

The learning rate of the self-organizing map determines how well the map can be trained, i.e. how precisely the structure of the input space can be captured. The effect can best be illustrated by an example: Figure 6-16 shows three document maps – each consisting of a 20×30 grid – of a collection of 28 documents (note that some input patterns have been mapped to the same unit during training). The maps have been trained with a different number of learning cycles per document. The different document symbols indicate the membership of each document to a given category. It can be seen that the global grouping of documents remains the same (the varying orientation of the map results from the random initialization of the neurons' weight vectors). However, the structure displayed by the left map is rather blurred compared to that from the middle. The right map's structures are worked out most precisely. Experience shows that 15 to 25 learning cycles per document are sufficient for a reasonable visualization of structures in the context of the document map approach.

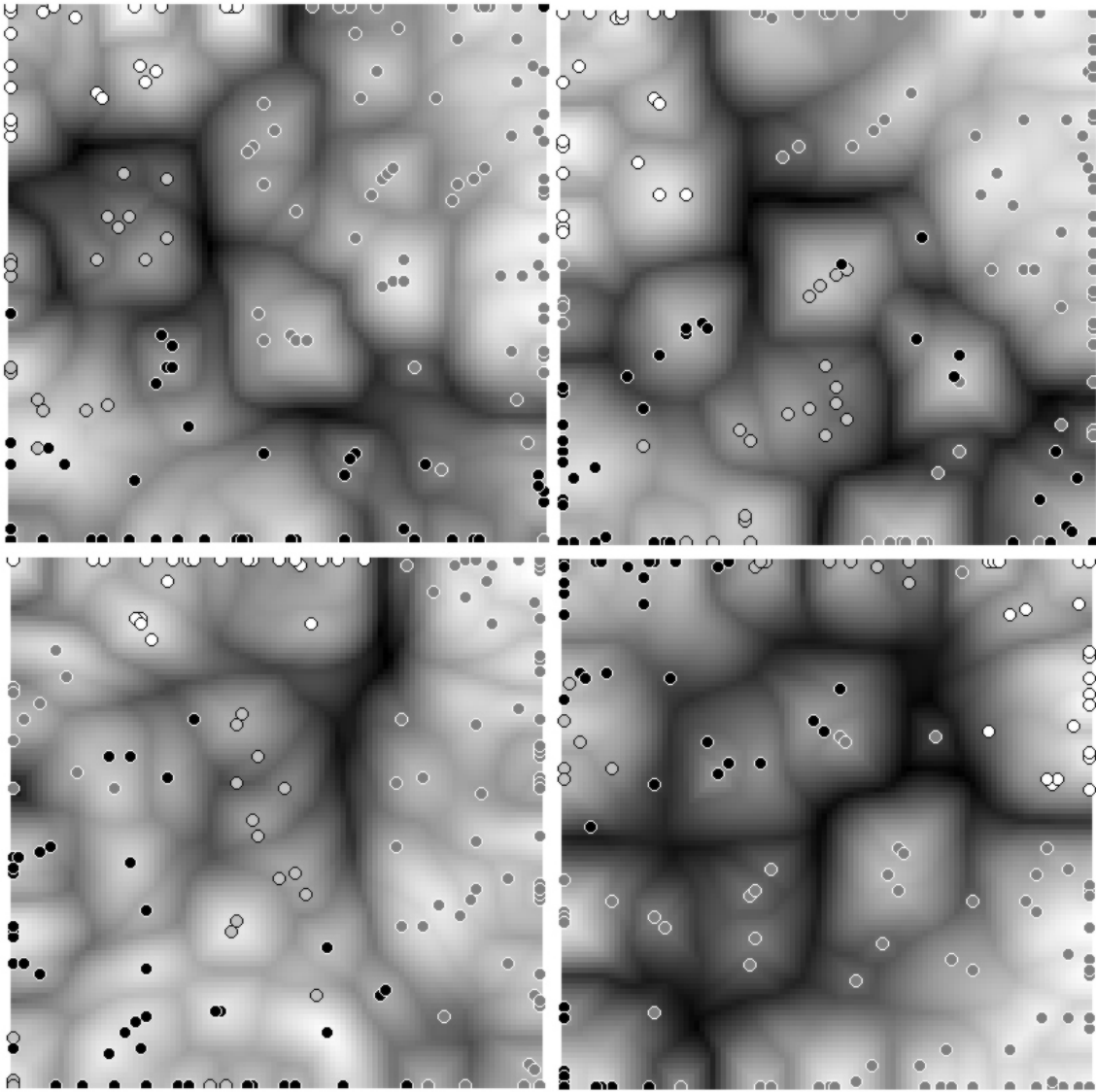


Figure 6-17: Document maps resulting from different similarity-to-distance transformations. From left to right and top to bottom: (a) linear, (b) cosine-based, (c) polynomial with $\lambda=0.5$, and (d) polynomial with $\lambda=2$

6.4.3 Influencing Structures by Weighting Document (Dis-)Similarities

In section 6.2.1.5 different similarity-to-dissimilarity transformations and distance weighting functions have been proposed. Such a transformation or weighting function is an important parameter of the basic framework since it allows to directly influence the visible structure of the document maps. In order to illustrate its effect figure 6-17 shows four document maps which visualize semantic document spaces that have been generated for the same collection of 168 documents but using different similarity-to-dissimilarity transformations. For the first semantic document space a linear similarity-to-dissimilarity transformation has been applied, the second was computed using a cosine-based transformation. The last two spaces have been constructed based on polynomial transformations. Each map has been calculated by a 100×100 SOM. The dimension of each document space on which the respective SOM is based has been chosen such that the stress values are similar (note that for the polynomial transformation with $\lambda=0.5$ the minimal stress value which can be achieved is 0.19). Table 6-2 gives the dimension and stress values for the different document spaces which formed the basis for the visualization.

The graphical document points in the map comprise four different symbols, each of which indicates a manually defined group to which the respective document belongs. In all four maps the grouping of documents is essentially the same: The white documents form a coherent, more or less separated group, the dark gray group has a coherent center but there is also some intermingling mainly with the black group and so on. However, the maps are significantly different regarding the visible structure. It can be seen that map (b) from figure 6-17 displays a stronger separation especially of the dark gray group than map (a). Map (c) does not show a clear separation at all whereas map (d) displays some strictly clustered groups. Considering map (b) it is nice to see how similar documents (regarding their group membership) are even closer together compared to map (a). Among others, the effect of different weighting and transformation functions gives rise to the question whether it is possible to measure the degree to which a given document space is structured.

Table 6-2: Dimensions and stress values for semantic document spaces based on different similarity-to-dissimilarity transformations

document space	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
transformation	linear	cosine-based	polynomial, $\lambda = 0.5$	polynomial, $\lambda = 2$
dimension	15	20	7	50
stress	0.09	0.1	0.19	0.1

6.4.4 Excursus: Measuring the Semantic Document Space's Structure

The last section has shown how the structure of the document space can be influenced by the different weighting functions for (dis-)similarity values. It is useful to examine the question whether there is a suitable measure which determines the 'degree of structuring' of the semantic document space. Such a measure could provide some additional basis for judging whether a given document space is weakly or rather strongly structured. What does 'degree of structuring' mean, actually? Intuitively, if the documents are distributed rather equally within the document space there is only a weak (or no) discrimination of document groups. In contrast, a heterogeneous distribution imposes a certain structure on the space and corresponds to a higher degree of structuring.

Provided that the grouping of documents is not distorted too heavy – i.e. that the semantic relationships are preserved on the whole – it would be good to have a rather clear and distinct structure so that information about document grouping can easily be derived from a given document map. A measure for the 'degree' of the semantic document space's structure would allow, for example, to compare the effect of different parameter settings which influence the document space's structure on a more objective level. Note, however, that ultimately the selection of parameters depends on the human analyst who has to judge the quality of document maps with respect to his interest and know-how. But still, a suitable measure would be of help for studying the impact of parameter adjustments. In this section an attempt for a suitable definition will be presented.

6.4.4.1 Frequency Distribution of Distance Values

The first thing that comes into mind when looking for the desired measure is considering the distribution of distance values within the given document space. This is because the inter-object distances 'induce' the structure of the document space. Intuitively, one can expect that in a rather homogeneous distribution of objects there is also a homogenous distribution of distance values, i.e. there are nearly as much large as medium or small distances. Distributions of distance values which deviate to a certain degree from such a homogenous distribution indicate a correspondingly stronger structure.

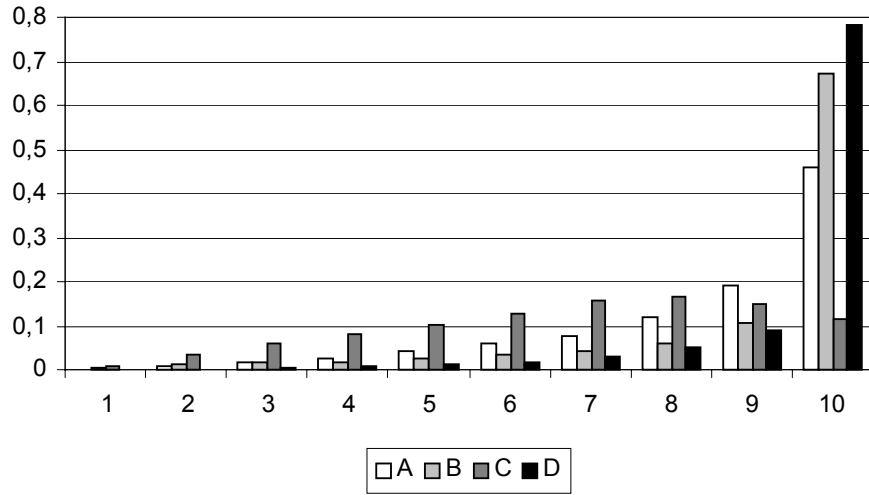


Figure 6-18: Distance distributions Δ^g with $g = 10$ for the document spaces from section 6.4.3

Based on this intuition a distance distribution function Δ can be defined which determines the relative frequency of distance values within a given interval and, thus, gives an idea of how the distance values are distributed within the considered document space. More precisely, let $d_{\min} =_{\text{def}} \min\{\delta^*(\mathbf{x}_j, \mathbf{x}_k) \mid \mathbf{x}_j, \mathbf{x}_k \in X, j \neq k\}$ denote the minimal distance between all pairs of document representatives from the semantic document space X (d_{\max} analogous). Let $[d_{\min}, d_{\max}]_g$ denote a partition of the interval $[d_{\min}, d_{\max}]$ into g disjoint intervals.

Formally, for $\varphi =_{\text{def}} (d_{\max} - d_{\min})/g$

$$[d_{\min}, d_{\max}]_g =_{\text{def}} [d_{\min}, d_{\min} + \varphi] \cup \bigcup_{i=1, \dots, g-1} [d_{\min} + i \cdot \varphi, d_{\min} + (i+1) \cdot \varphi]. \quad (6-22)$$

Then, for $j \neq k$, $1 \leq j, k \leq n$, $\mathbf{x}_j, \mathbf{x}_k \in X$, the distance distribution Δ^g with granularity g is a function $\Delta^g: \{1, \dots, g\} \rightarrow [0, 1]$ defined as

$$\Delta^g(i) =_{\text{def}} \frac{|\{(\mathbf{x}_j, \mathbf{x}_k) \mid \delta^*(\mathbf{x}_j, \mathbf{x}_k) \in [d_{\min}, d_{\min} + \varphi]\}|}{(|X|^2 - |X|)}, \quad (6-23)$$

$$\Delta^g(i+1) =_{\text{def}} \frac{|\{(\mathbf{x}_j, \mathbf{x}_k) \mid \delta^*(\mathbf{x}_j, \mathbf{x}_k) \in [d_{\min} + i \cdot \varphi, d_{\min} + (i+1) \cdot \varphi]\}|}{(|X|^2 - |X|)},$$

for $1 \leq i \leq g-1$. Obviously, $\sum_i \Delta^g(i) = 1$.

Figure 6-18 shows four distance distributions Δ^g with granularity $g = 10$ for the document spaces discussed in section 6.4.3. The distance distribution for semantic document space C (cf. table 6-2) shows the weakest deviation from a homogeneous distribution, followed by the distributions for A , B and D . This observation corresponds to the visual impression of the degree of structuring as conveyed by the corresponding document maps (cf. figure 6-17).

6.4.4.2 Distribution Factor

Besides considering the frequency distribution of distance values, is there a suitable measure which yields a single value for the ‘degree of structuring’ of the semantic document space? The idea is to define a measure, called ‘distribution factor’, which indicates the degree to which the document representatives are scattered among the semantic document space. This

can be done by measuring the *information content* of a suitable distribution function based on results from information theory. Prior to the definition of the distribution factor some background on the notion of entropy shall be given.

In his ‘source coding theorem’ Claude Shannon motivates entropy as a measure for the information content of a discrete data source [Shan48]. A discrete *data source* or *channel* produces messages by sending output symbols of a finite alphabet in discrete time instants [Blah87]. Formally, it consists of an output alphabet $A = \{a_0, \dots, a_{n-1}\}$ and a probability distribution $\mathbf{p} = \{p_j\}$ on A , where $p_j =_{\text{def}} p(a_j)$, $a_j \in A$, $p_j \geq 0$, $\sum p_j = 1$. The probability distribution \mathbf{p} models the behavior of the data source. The information content or entropy H of a discrete data source \mathbf{p} is defined as

$$H(\mathbf{p}) =_{\text{def}} - \sum_{j=0}^{n-1} p_j \log_2(p_j) = \sum_{j=0}^{n-1} p_j \log_2 \left(\frac{1}{p_j} \right). \quad (6-24)$$

$H(\mathbf{p})$ characterizes the degree of uncertainty about the outcome of \mathbf{p} . A selection of some basic properties of H (without proofs) underlines this intuition (for a complete discussion of the properties of H cf. [Blah87]):

- (1) The entropy function H is continuous in \mathbf{p} , i.e. small changes in \mathbf{p} cause only small changes in $H(\mathbf{p})$. This is reasonable since a slight change in the behavior of a data source only slightly changes the uncertainty of its outcome.
- (2) $H(\mathbf{p}) \geq 0$ and $H(\mathbf{p}) = 0$ if and only if $p_i = 1$ for some i . In that case there is no uncertainty about the outcome of \mathbf{p} and, thus, no information content.
- (3) $H(\mathbf{p}) \leq \log n$. If $p_i = 1/n$ for all i then $H(\mathbf{p}) = \log n$. This is because the uncertainty is largest if all outcomes have the same probability. Furthermore, in this case H is a monotonously increasing function of n . This means that the uncertainty increases with the number of possible messages to choose from.

The notion of entropy can be used as an auxiliary to measure the degree of structuring of the semantic document space. The only question is how to choose a suitable probability distribution \mathbf{p} . Obviously, the frequency distribution of distance values Δ^g is a possible candidate. Since a homogeneous distribution of distance values corresponds to a homogeneous scattering of objects in the document space a high entropy of Δ^g corresponds to a low ‘degree of structuring’ and vice versa.

However, given any three objects $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c$ from the semantic document space, their respective distances are not independent due to the fact that the triangular inequality holds for δ^* on X . Thus, the ‘probabilities’ measured by Δ^g are not independent. Despite that fact, entropy can help to assess the information content of a single distance distribution and allows a ranking of a given set of distance distributions. The *distribution factor* $f(\Delta^g) \in [0,1]$ for a distance distribution Δ^g can be defined as

$$f(\Delta^g) =_{\text{def}} \frac{1}{\log_2 g} \cdot \sum_{i=1}^g \Delta^g(i) \cdot \log_2 \left(\frac{1}{\Delta^g(i)} \right). \quad (6-25)$$

If $\Delta^g(i) = 0$ then the respective product in the numerator is set to zero. High values of $f(\Delta^g)$ correspond to a high degree of scattering, i.e. a low degree of structuring. Table 6-3 shows the distribution factors for the document spaces from section 6.4.3. The values pretty well express the intuitive impression of the degree of scattering of each document space: The map of space C shows a rather homogeneous distribution of documents whereas D ’s map is relatively strongly structured. The map for A conveys a certain structure but the map for B is signifi-

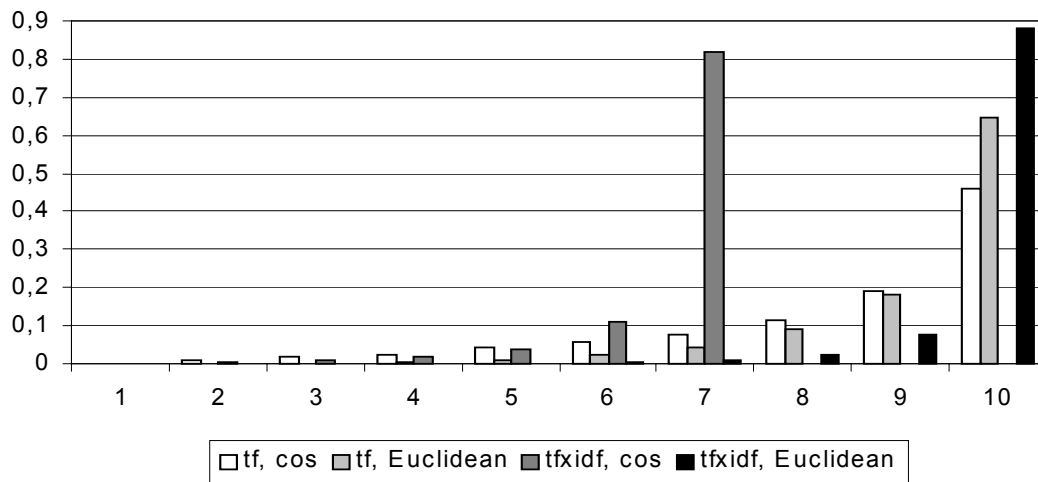


Figure 6-19: Distance distributions with $g = 10$ for the document spaces from table 6-4

cantly stronger structured. Note, however, that $f(\Delta^g)$ is not a measure of the space's quality in some sense. In particular, $f(\Delta^g) = 0$ does not indicate an 'optimal structure' but only the strongest structure with respect to g .

Table 6-3: Distribution factors for different semantic document spaces

document space	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
transformation	linear	cosine-based	polynomial, $\lambda = 0.5$	polynomial, $\lambda = 2$
dimension	15	20	7	50
$f(\Delta^g)$	0.71	0.55	0.93	0.38

To conclude, the proposed distance distribution and the distribution factor are reasonable measures which convey a certain picture of the degree to which a given semantic document space is structured. However, further approaches would be possible. For example a different probability distribution for the entropy-based distribution factor could be chosen. An alternative would be to divide the given document space into discrete, equally-sized cells and count the relative frequency of document representatives falling into each cell. Whereas this would avoid the theoretical disadvantage of non-independent probabilities it produces some other disadvantages. The most significant one is that the number of cells exponentially grows with the number of dimensions of the document space. As a consequence most cells will contain zero or only one document representative. Thus, this approach is only feasible for a small number of dimensions. A deeper discussion of possible measures would be beyond the scope of this work.

6.4.5 Parameters of the Document Analysis Module

Since in the proposed framework the document analysis module can be exchanged the parameters to be set depend on the concrete realization. In this section the settings of the default analysis module based on the vector space model will be discussed. Basically, there are two parameters which can be varied: the way term vector components are weighted, i.e. the choice of the local, global and normalization component of the term weights (cf. equation (6-1) in section 6.2.1.3), and the document matching function. In this realization of the vector space model the normalized term frequency (*tf*) and the *tfidf* weighting scheme as well as the Euclidean distance and cosine similarity as document matching functions are used. Table 6-4 presents the four possible combinations of weighting scheme and document matching func-

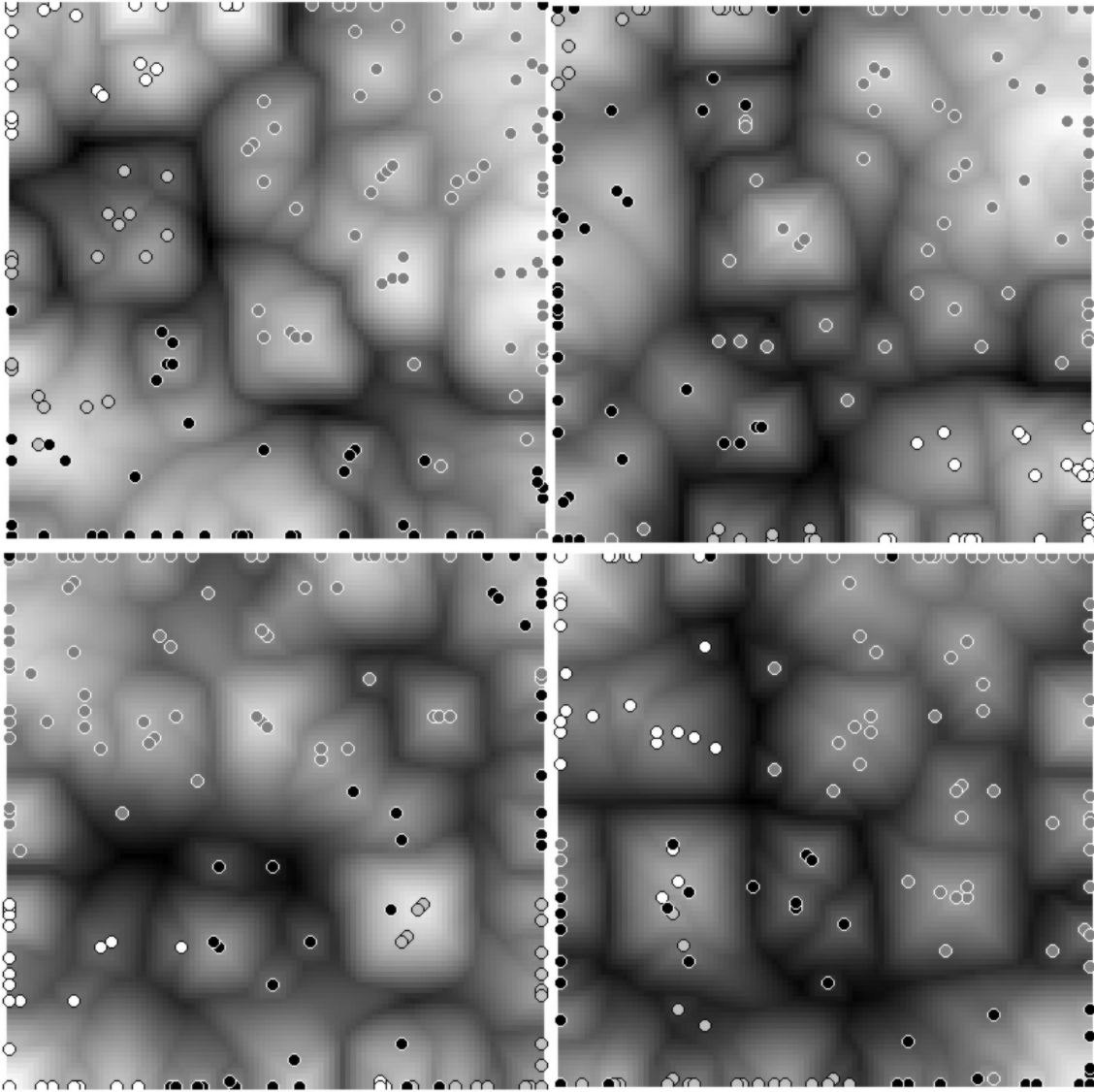


Figure 6-20: Document maps resulting from different term vector weightings and dissimilarity measures. From left to right and top to bottom: (a) *tf*, cosine, (b) *tf* Euclidean, (c) *tf* \times *idf* cosine, and (d) *tf* \times *idf* Euclidean

tion. For each combination and a fixed collection of 168 documents (cf. section 6.4.3) a semantic document space has been computed and visualized (cf. figure 6-20). The number of dimensions for each space has been chosen such that the resulting stress values for the different spaces are comparable. Moreover, for each document space the table shows the distribution factor (the corresponding distance distribution is shown in figure 6-19) which underlines the impression of the degree of structuring conveyed by the maps.

Table 6-4: Parameters and statistics of the document analysis module and the semantic document space

term weighting, document matching	dimensions	stress	distribution factor
<i>tf</i> , cosine (linear conversion)	15	0.09	0.71
<i>tf</i> , Euclidean	60	0.1	0.48
<i>tf</i> \times <i>idf</i> , cosine (linear conversion)	45	0.09	0.29
<i>tf</i> \times <i>idf</i> , Euclidean	90	0.1	0.21

Again, the document points are color-coded differently according to a given grouping. Though the global neighborhood relations of documents remain the same in all maps the degree of structuring is significantly different. Figure 6-20 (a) rather faithfully reflects the coarse group structure and displays the overlapping of some groups. The same is true for

figure 6-20 (b) but this time there is a significantly finer structuring among the subgroups (cf. especially the dark gray and black group). In contrast, in figure 6-20 (c) and (d) it is harder to identify the given main groups since the maps are divided into relatively small and demarcated sectors.

First, the difference between maps in the top and the bottom row of figure 6-20 shall be discussed. The bottom row shows distinctly clearer structures of subgroups whereas the coherence of the main groups is harder to identify. The documents are compared regarding the extracted indexing terms. The main groups in this document collection are characterized by certain key terms which thus appear relatively frequently in the whole collection. Consequently, the global *idf* weight component of the *tf×idf* measure will be rather low and decrease the weight of the corresponding term. The result is that the main groups do not appear as strictly separated sectors in the map. Collections where relatively large groups are characterized by certain key terms can be found quite often in the domain of technical text collections. Regarding the document matching function, there is also a difference between the left and the right column of figure 6-20: The ‘cosine’ maps on the left show larger groups than the ‘Euclidean’ maps on the right. This is due to the nature of the measures: For the cosine similarity only terms which are shared by a pair of documents influence the result whereas the Euclidean measure takes into account all terms of the document pair.

Summing up, the sample maps presented here illustrate the effect of different weighting and matching function combinations for a given collection of documents. However, it is not possible to identify a ‘best weight and measure’ combination since the choice depends on the collection type and the task which shall be performed with the map. Which combination of measures is best suitable is ultimately a matter of exploration. However, this section has given some hints regarding the effects of the parameter setting on the resulting document map.

6.4.6 Complexity of the Document Map Approach

A few notes on the complexity of the document map approach are appropriate. This section focuses on the complexity of the spatial scaling module (i.e. its realization in this work) and the topology preserving mapping and visualization. The complexity of the document analysis module depends on the chosen realization. Once computed the internal document representations can be used for the generation of multiple maps. Anyway, in most applications the complexity of spatial scaling and topology preserving mapping will dominate.

Given a collection of n documents the computation of the dissimilarity matrix would require $O(n^2)$ document comparisons. Depending on the complexity and design of the spatial scaling method it is not necessary to compute the complete matrix *a priori*. Rather, the dissimilarity for pairs of documents can be computed on-the-fly when required by the scaling method. The scaling method used in this work has a complexity of $O(m \cdot n)$ where m denotes the desired number of dimensions of the semantic document space. Obviously, not all the dissimilarity data available is required by this method. The complexity is true for both, the original and the modified version of the scaling method (cf. section 6.2.2). However, if it is desired to convert the dissimilarity data into metrical distance information the dissimilarity matrix has to be computed first. The data can then be transformed into a metric in time $O(n^3)$ or prepared in time $O(n^2)$ using the heuristic from section 6.2.2.2.

The dominant part of the computation is the training of the self-organizing feature map. Given a $k \times l$ grid structure, let s_{net} denote the number of units in the grid, i.e. $s_{\text{net}} = k \cdot l$. Determining the cluster center for a given input pattern requires time $O(m \cdot s_{\text{net}})$ since for every unit the Euclidean distance between its weight vector and the given input pattern is calculated (the calculation of the distance can be done in time $O(m)$). The following weight adjustment again

Table 6-5: Computational effort for different document collections and parameter settings

document collection	# documents	60	168	679	929	2000	4770
	avg. number of words	152	140	225	150	258	160
	avg. # of indexing terms	38	29	42	33	70	37
	size of vocabulary	632	806	2323	2,809	24,249	10,273
document map	dimension of document space	20	20	75	120	100	150
	size of SOM	30×30	50×50	100×80	100×100	140×100	150×250
	training cycles per document	25	20	20	20	20	15
computation time	indexing	0.62 sec	1.58 sec	10.83 sec	11.04 sec	19.78 min	1.53 min
	similarity and spatial scaling	0.09 sec	0.27 sec	5.97 sec	8.92 sec	1.33 min	3.02 min
	SOM training, visualization	2.73 sec	23.92 sec	23.7 min	1.03 h	2.6 h	19.18 h
	total time of computation	3.44 sec	25.77 sec	23.98 min	1.03 h	2.95 h	19.26 h

needs time $O(m \cdot s_{\text{net}})$. Consequently, if tc denotes the number of training cycles per document (more precisely, $t_{\text{max}} =_{\text{def}} tc \cdot n$ in algorithm 6-3) the complete computational effort is in $O(tc \cdot n \cdot m \cdot s_{\text{net}})$. Finally, the extraction of the network's topological information according to section 6.2.3.2 requires time $O(n \cdot m \cdot s_{\text{net}})$ which can be easily checked by considering equation (6-20). According to experience reported in section 6.4.2, the ideal size s_{net} of the grid should be in the range of $5 \cdot n \leq s_{\text{net}} \leq 10 \cdot n$, so that $s_{\text{net}} \in O(n)$, and the number of training cycles tc is usually $15 \leq tc \leq 25$, thus $tc \in O(1)$. Then, the complexity estimation for training the network as well as extracting the topological information reduces to $O(n^2 \cdot m)$.

Table 6-5 gives an overview of the computational effort for different document collections and parameter settings. For all experiments the documents have been indexed by the vector space model using tf weights and the cosine measure of similarity. The experiments have been performed on a Pentium II 350 MHz machine with 128 MB RAM and WinNT 4.0. Note that the prototype document map system is not optimized regarding calculation time, i.e. there is some space for improvement though the overall complexity will, of course, remain the same.

To conclude, the computation of a document map requires a non-trivial computational effort. However, the map generation itself is an offline procedure; interaction with the map can be done in real time (cf. chapter 7).

6.5 Experiment: Structuring Newsgroup Articles

Evaluation of document maps is a difficult topic since there is no accepted quantitative measure for the quality of such a map. The aim of the proposed approach is to display the similarity structure of a collection, not to categorize documents. Thus, classic information retrieval measures like precision and recall do not apply here since they require a clearly defined 're-

Table 6-6: Categorization of 20 newsgroups

social affairs	computer	recreation	science	miscellaneous
religion	windows	automobiles	CS related	for sale
alt.atheism soc.religion.- christian talk.religion.misc	comp.os.ms- windows.misc comp.windows.x	rec.autos rec.motorcycles	sci.crypt sci.electronics	misc.forsale
politics	hardware	sport	medicine	
talk.politics.guns talk.politics.- mideast talk.politics.misc	comp.sys.ibm.- pc.hardware comp.sys.mac.- hardware	rec.sport.baseball rec.sport.hockey	sci.med	
	graphics		space	
	comp.graphics		sci.space	
600 articles (30%)	500 articles (25%)	400 articles (20%)	400 articles (20%)	100 articles (5%)

sult set' of documents which can be examined (see also [MHNW97]). Furthermore, the notion of 'relevance' would have to be defined appropriately – but with respect to what? Artificial and unsuitable measures would disguise the quality rather than assess it. Regarding the document map *technology*, there are several points to apply the lever for quantitatively measuring a map's quality: It is possible to measure precision and recall of the applied document comparison module itself and to assess the information loss when mapping documents regarding their dissimilarity into a high-dimensional space (cf. chapter 6.4). Furthermore, there are approaches (which are, however, not yet commonly accepted) for measuring the degree of topology preservation of the SOM (cf. [BaPa92]). Finally, the visualization technique shows the information contained in the trained network with respect to the clusters in the input space. A discussion of this method can be found in [Sklo96]. However, it is not possible to draw conclusions for the overall approach from the quality of its single components.

In this section the overall document map approach is applied to a test collections of newsgroup articles for which a classification of documents is available. Since documents belonging to common classes are supposed to be semantically related to one another to some degree, it can be checked whether the document map reflects parts of the class structure. Note, however, that it is not reasonable to strictly compare the given class structure with the structure shown by a document map since there may be inter-class relationships which cause meaningful similarities. Therefore, the visible structures are discussed regarding their plausibility. Moreover, the scalability of the approach is examined. A more detailed evaluation than presented in this section would have to include an application context and domain experts – a task which will be tackled in chapter 8 which deals with extensive case studies.

In this experiment the document map approach is applied to a standardized and commonly available test collection of newsgroups [Mit99]. Though the nature of the collection does not correspond to the intended application domain of the proposed document map technology it is interesting to check whether the approach is able to separate groups of related documents in a heterogeneous collection. Moreover, the scalability of the method shall be examined.

The collection consists of 2,000 messages taken from 20 different newsgroups. From each newsgroup 100 Usenet articles were taken. To simplify the graphical overview these newsgroups have been categorized according to their general topic. The categories and respective groups are shown in table 6-6. Regarding the data characteristics the articles are typical postings. The original data as provided by the donor includes subject lines, signature files, and quoted portions of other articles. Divergent from that all header data has been removed apart from the subject line to avoid similarity artefacts based on header information such as the

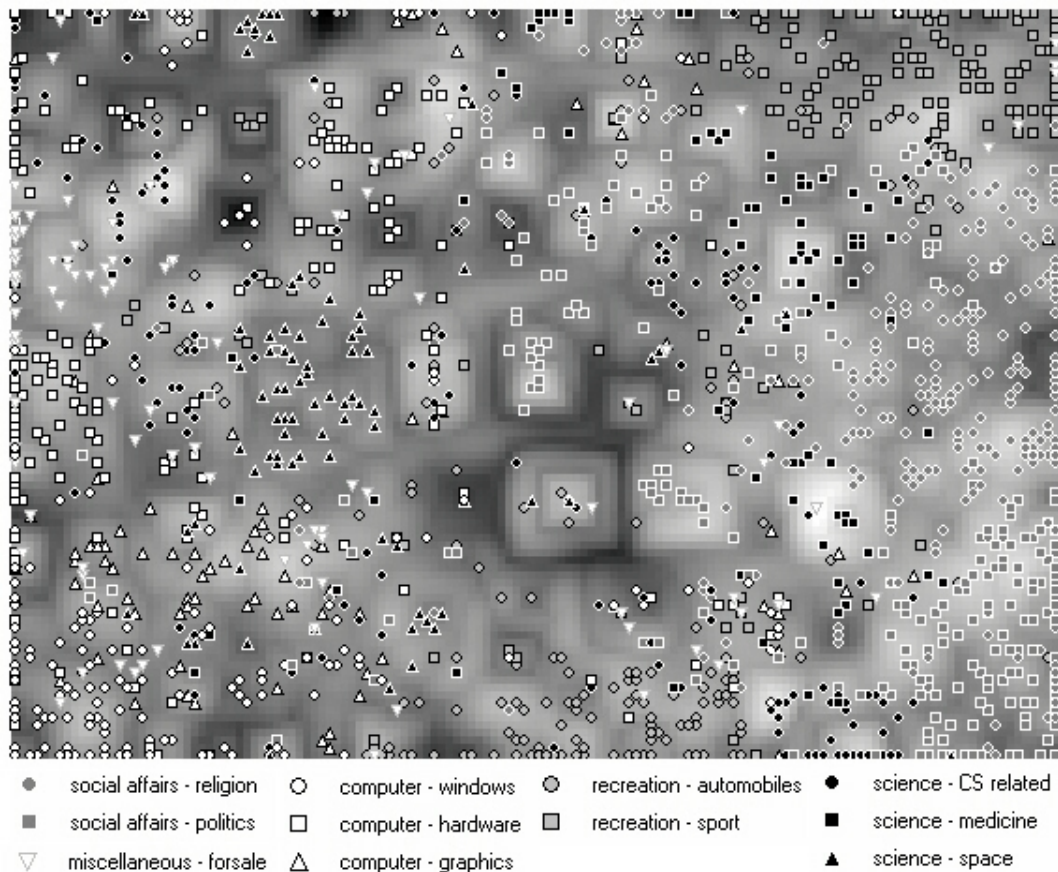


Figure 6-21: Map of 2,000 Usenet articles

name of the newsgroups in which an article has been posted. Note that approximately 4% of the articles are crossposted.

A document map has been generated according to the parameters given in table 6-7. Looking at the resulting map (figure 6-21), there is a clear tendency that documents which belong to the same category are to be found in common areas, thus forming coherent groups. In more detail, most articles related to “social affairs” are coherently located at the south-eastern edge of the map. Some postings (approximately 17%), mainly belonging to groups about politics, are scattered in the central part. A few documents can be found even in the south-western area. “Computer”-related newsgroup articles are concentrated in the western and south-western part. Again, many articles from each group form a main cluster whereas others are scattered around the main area. Articles about automobiles from the “recreation”-group are clearly concentrated in the mid-southern part with some scattered documents to be found around the central part. A similar situation can be observed with the “sport” documents from the “recreation”-category. They are located mainly in the north-eastern corner, less than 20% are scattered in the central and north-western area. The “science” groups form less differentiated clusters. Although three main clusters can be identified in the mid-western (“space”), mid-eastern (“medicine”) and south-eastern part (“computer science”) a significant amount of documents is scattered around the central areas. Finally, roughly one third of the “for sale”-articles are clustered at the north-western edge, the remaining documents can be found near postings belonging to the “computer”, “sport” and “automobiles” categories.

The overall group relationships revealed by the map can be judged as meaningful in general, although there are clearly some semantically unjustified similarities due to the simple document comparison method applied here. Note that the aim of this document map method is not to categorize documents (in particular it is not intended to learn and rebuild the given classifi-

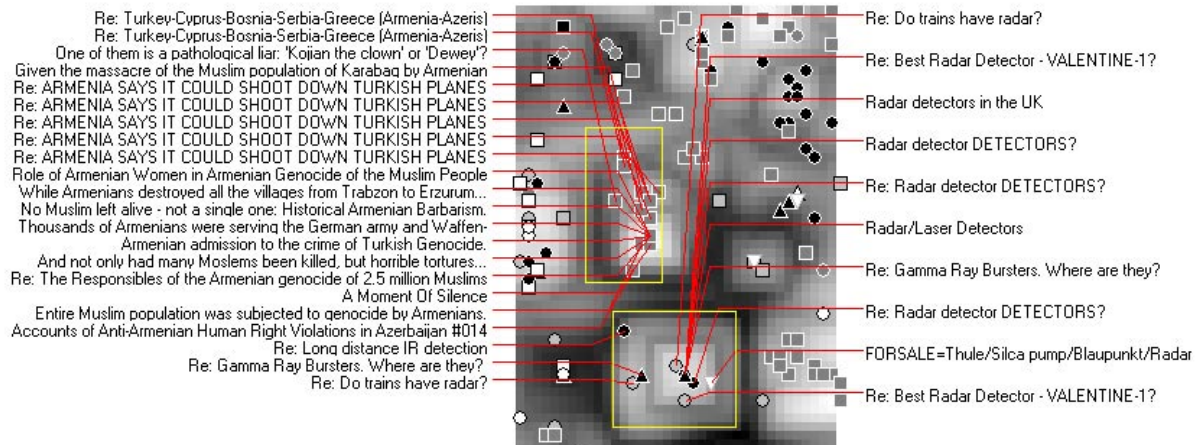


Figure 6-22: Detail of '20 newsgroups' map

cation) but to show their relationships and to visualize the connection between groups and categories. The newsgroup map contains interesting information about inter-group relationships. For example, consider figure 6-22: The document group at the bottom of the map contains postings from different newsgroups. However, the messages share a common topic, namely 'radar devices'. Additional newsgroup maps with up to 10,000 documents are shown in appendix A. All these maps contain rich information about the collections' structure. However, the more documents are depicted, the harder it becomes to derive structural information from the color distribution of the visualization alone. Rather, relationships between pre-defined classes can be studied effectively. An open question is the usefulness of the map approach in the intended application domain of managing the knowledge contained in specialized document collections – a research question which is addressed in chapter 8 and 9.

Table 6-7: Parameters and statistics of collection and map. Concerning the given computation tim note that all experiments have been performed on a Pentium II 350 MHz machine with 128 MB RAM and WinNT 4.0.

collection	name	20 newsgroups
	# documents	2,000
	avg. # words	258
indexing	size of vocabulary	24,249
	avg. # indexing terms / document	70
	weighting scheme	TF
	similarity measure	Euclidean
doc. space	dimension	100
	avg. stress	0.75
SOM	size of SOM	140 × 100
	training steps per document	20
	total computation time	3 h 1 min

7 DocMINER – An Interactive Map-Centered Corpus Analysis Tool

This chapter presents the prototypical realization of the method proposed in chapter 6 and its incorporation into an interactive system environment. Section 7.1 briefly discusses the objective of developing the system, section 7.2 sketches the system's architecture and its connection to the basic framework for generating document maps. After that, sections 7.3 and 7.4 present the graphical user interface and introduce additional interactive features along with the methods behind them. Section 7.5 discusses usability issues before a sample document map session using the system is presented (7.6).

7.1 Objectives of the System Development

In this work design and use of document maps for supporting corpus analysis tasks in knowledge management are studied, driven by the requirement to allow pattern discovery and supporting the examination of document relationships (both, inherent and according to external specifications, cf. section 2.6.1). Document maps as such are a passive graphical presentation of a text collection's similarity structure. Corpus analysis tasks, however, require an interactive access to a document collection. Thus it is necessary to strive for a suitable document map system.

How should such a system be designed? The task model in chapter 2 already suggests functions for getting a quick map overview on different levels of detail, connecting goal-directed and explorative text access and incorporating meta-information like category membership. What other arguments for design issues are relevant? Shneiderman [Shn96] proposes the 'Visual Information Seeking Mantra' as a useful starting point for designing advanced visual user interfaces: Overview first, zoom and filter, then details-on-demand – a principle which can be discovered in many information visualization projects. More precisely, Shneiderman identifies seven tasks – overview, zoom, filter, details-on-demand, relate, history and extract – that need to be supported by advanced graphical approaches (cf. chapter 2.3.2 for more details on this task model). According to [Hea99], information access tasks “*span the spectrum from asking specific questions to exhaustively researching a topic*” (p. 262). In particular, if corpus analysis is seen as a general process which involves explorative as well as goal-directed phases (cf. chapter 2.6), the overall goal of tool development must be to support both ways of accessing information. The document map itself provides an overview of the inherent similarity structure of a text corpus. It is now necessary to provide suitable means for getting detail information, connecting the graphical display with goal-directed search types, and enabling to enrich the map with user defined meta-information. Experiences in the field of document map systems show that a tight integration of querying and browsing would help to improve the usefulness of existing approaches (cf. [Hea99] and [Lin95]). This hypothesis is also supported by [HeFr96] who found out that users like an integration of scanning and querying in system interfaces.

These considerations have led to the development of a prototypical document map system: DocMINER (Document Maps for Information Elicitation and Retrieval) is an interactive, map-centered corpus analysis and text-mining tool which tightly integrates the map display with explorative and goal-directed interaction methods. It has been developed in order to study the applicability and to evaluate the usefulness of the approach in knowledge management. In particular, it was used for conducting case studies in real-world contexts and for examining the value added by visualizing the similarity structure of a text corpus for performing typical analysis tasks in a more formal setting. But moreover, it is a contribution of its own since it experiments with interactive functions which are tightly coupled with the graphical presentation of the document map, intending to support analysts to grasp the structure of the document space.

7.2 Overall Architecture

Before the user interface of DocMINER will be presented, this section gives a general view on the architecture of the prototypical system which reflects the modular design of the basic framework. Figure 7-1 depicts the main modules of the system, i.e. the components which are concerned with generating and interacting with a document map; control modules or control flows are not shown. The system's graphical user interface and the map calculation engine are separate system modules. The latter is technically realized as a dynamic link library. The core of the system is the structuring engine. Its main components are

- the variable document analysis module according to the basic framework (in the developed prototype the document analysis module is a realization of the vector space model, cf. chapter 6.2.1),

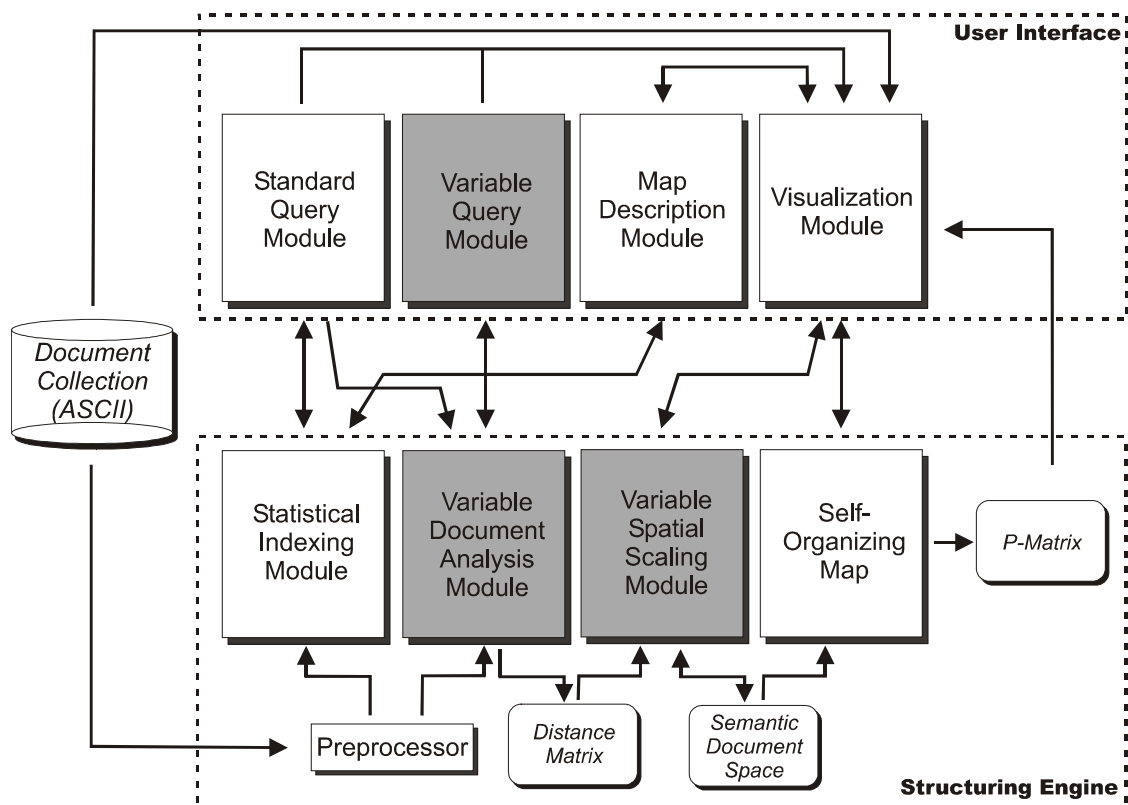


Figure 7-1: General architecture of the document map system. The colored modules of the structuring engine are variable in the basic framework. The (optional) variable query module of the user interface has to be chosen w.r.t. the document analysis module.

- the variable spatial scaling module as required by the basic framework (realized in the prototype as proposed in chapter 6.2.2),
- a module realizing the self-organizing feature map which actually computes the document map and the visualization information (P -matrix cf. chapter 6.2.3), and
- a statistical document indexing module which is necessary for supporting standard queries and advanced document information functions (a realization of the vector space model, cf. sections 7.4.2 and 7.4.3).

Each document of the collection is stored as a single text-file on disk. These files may contain markup tags which describe the section-structure of the document (title, headings, abstract, body, etc.). According to a structure definition file, the preprocessor selects only those parts of the documents which are indicated as relevant for being indexed and hands them over to the statistical indexing as well as the document analysis module.

The graphical user interface communicates with the modules of the structuring engine. The visualization module displays the topological document space information provided by the P -matrix and interacts with the spatial scaling module and the self-organizing map in order to place document points on the map display according to their position in the semantic document space. The map description module uses information provided by the statistical indexing module for calculating certain document group characterizations, some of which are also presented in the map display (cf. section 7.4.2). Furthermore, it manages user-defined map and document meta-information which can also be visualized in part (cf. section 7.4.1). The standard query module matches queries against the statistical document representations or interprets a query as a new document to be added to the map (cf. section 7.4.3). The query result is also displayed in the document map. Finally, it is possible to integrate a special query module which matches queries against the document representation of the variable document analysis module, thus exploiting the possibly more powerful domain-tailored document comparison and query matching method (according to the philosophy of the basic framework). Note that the class structure of the object-oriented implementation (realized in C++) allows to easily exchange the modules that are variable in the basic framework. A more consequent architecture would have been to realize the exchangeable modules in separate dynamic link libraries, but due to the prototypical character of the tool this step has not been taken.

7.3 Generating Document Maps: Project Workbench

This section sketches the procedure for generating document maps using the prototypical document map system DocMINER. Figure 7-2 shows the system's project workbench along with some parameter forms. Each DocMINER session starts with opening an existing or defining a new project using the project manager. For setting up a new project the user specifies a project name, a document collection, a stop word list containing general as well as domain specific common words, and optionally a domain specific thesaurus. The latter may contain sets of synonyms and a list of compound words. Stop words and thesaurus are used by the statistical document indexing and the standard query module⁴. Once a project is defined, different document maps can be generated for the considered collection. The project workbench of DocMINER shows a tree view of existing basic maps along with their parameters, annotated and color-coded maps produced for corpus analysis, semantically refined maps (see also chapter 10.9), and maps of certain sub-collections.

⁴ Phrase control is not implemented in the statistical document analysis module.

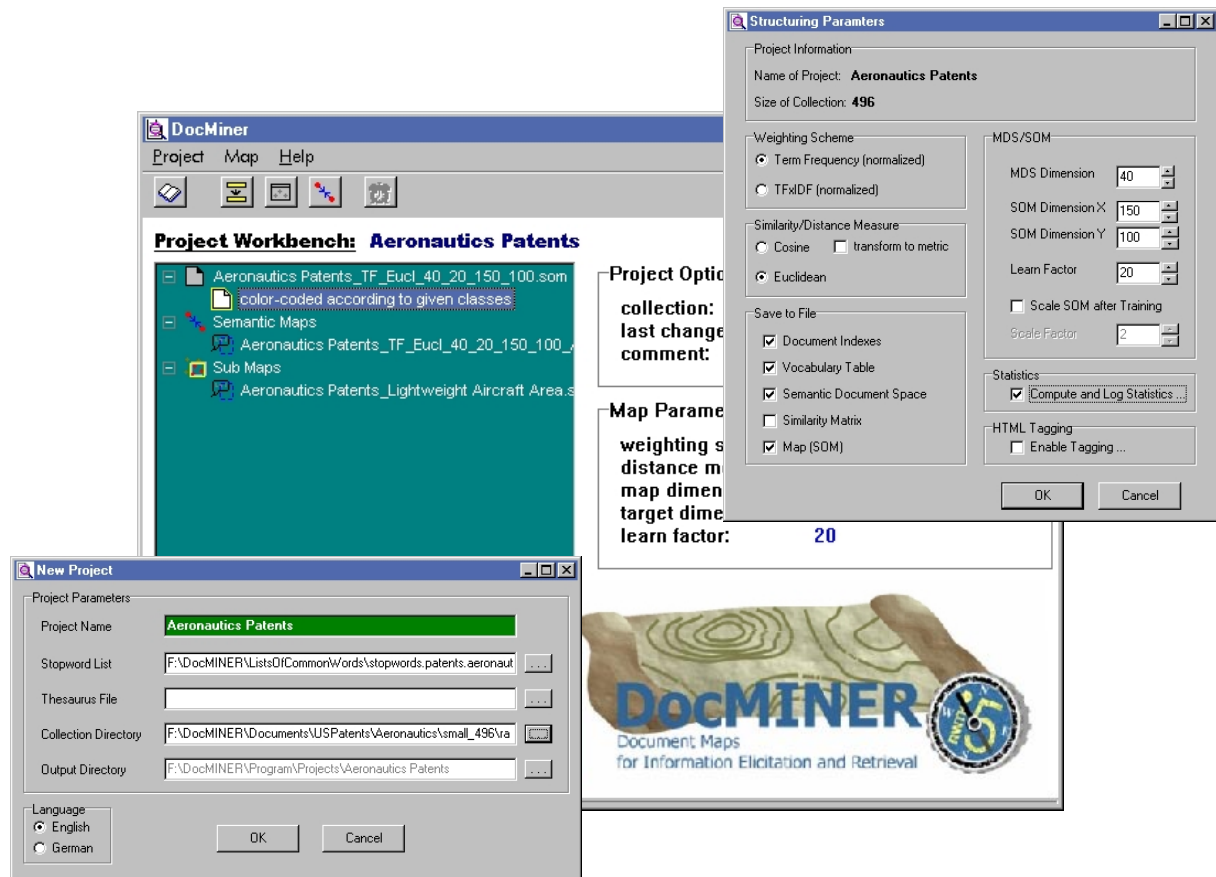


Figure 7-2: DocMINER's project workbench with a tree view of available document maps, definition forms 'new project' and 'structuring parameters'

Basic document maps can be computed according to the realization of the basic framework (chapter 6.2). Therefore, the structuring parameters form requires the specification of a weighting scheme (*tf* or *tf \times idf*), a document matching function (cosine similarity or Euclidean distance), the target dimension of the semantic document space, and the parameters for the neural network (size, training steps) for each map to be computed. Besides, many different statistics can be calculated, e.g. statistics on indexing and term distribution, on the stress development and structure of the semantic document space (cf. chapters 6.4.1 and 6.4.4), or on mapping mismatch and topographic product [BaPa92] of the self-organizing feature map. A basic map can be enriched with meta-information, comments and color-coding by the analyst (for details cf. section 7.4).

7.4 Working with a Map: Interactive Features of the System

As a visual presentation of a text collection's similarity structure a document map is passive in nature. In order to support a fruitful step-by-step exploration of the text corpus, the system DocMINER provides several interactive features which allow to gain information on the detected document groups on different levels of granularity. The philosophy of the tool is to tightly integrate the graphical map display with collection-centered interaction methods such as querying, assigning categories to documents or extracting terms and indicative sentences from texts. Consequently, the system consists of two parts, coupled by a bi-directional control and information flow: The document map as a visual workspace provides an overview of the text corpus and allows to select arbitrary documents and document groups for applying further

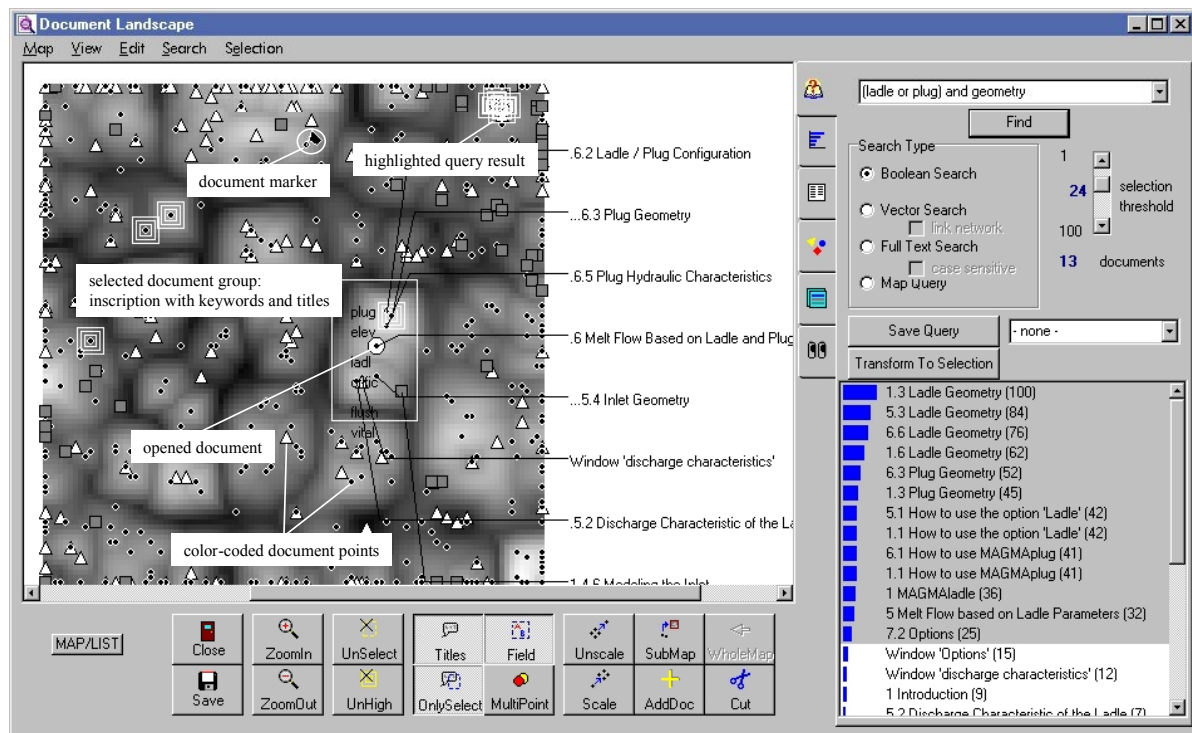


Figure 7-3: GUI of the system DocMINER. Document map workspace and base interface with registers (top to bottom: query, term profile, full-text, interactive legend, important sentences, interactive title list).

analysis and access methods. The base retrieval and access interface allows to pose queries, to directly select and access documents, and to extract and display information on selected texts. In return, retrieval and analysis results are not only displayed in separate windows but also in the visual map workspace whenever this is possible and useful. Figure 7-3 presents the user interface of DocMINER. The remainder of this section describes the interactive features of the system and, where necessary, the methods behind them.

7.4.1 Basic Document Map Interaction

7.4.1.1 Basic Functions of the Visual Workspace

Documents, represented as colored points in the map display, can be opened by point-and-click. Some document points may represent multiple documents due to a very high similarity of the corresponding texts. These multiple assignments can be highlighted in the map display. The text of an opened document is displayed in the full-text register (which shows one full-text at each time), but can also be shown in a separate window, thus allowing to display multiple documents. Opened documents are highlighted in the visual map workspace for easy navigation. The analyst can select arbitrary subsets of documents for further processing by drawing selection frames.

The titles of selected documents or the entire collection are displayed in the title register. This register also allows to open documents by clicking on their titles, to select documents by their names, and to assign user-defined categories to texts (cf. section 7.4.1.2). Besides using the title register as a compact list of selected documents, document points can be inscribed by the titles of the corresponding texts – a function which is useful for closely examining a rather small subgroup of documents.

DocMINER comprises a simple annotation function: Comments can be attached to documents, titles to be displayed can be changed, and markers for documents can be set. All in-

formation added to a document map (e.g. markers, annotations, color-coding) can be saved for future reference.

7.4.1.2 Color-Coding and Interactive Legend

DocMINER allows to define individual graphical symbols (combinations of different forms and colors) for each document point in the visual workspace. The ‘semantics’ of each symbol can be defined in an interactive legend. Once symbols are available, there are two major ways for enriching the map display by this meta-information. For one, the similarity structure of the collection as determined by the document analysis module can be superimposed by an externally defined document grouping. This allows, for example, to study the relationship of pre-defined groups with respect to the documents’ inherent similarity. On the other hand, this function allows to define categories for documents during the analysis of the map. In order to ease the overview, the analysts can highlight documents which are color-coded regarding a certain category by clicking on the symbol in the interactive legend.

7.4.1.3 Inspecting Details of the Map: Zooming, Scaling and Sub-Maps

DocMINER supports different functions for enabling a detailed analysis of certain map areas, the simplest of which is a standard zoom of the visual presentation. In order to preserve an overview, a miniature map can be displayed, marking the particular area of the entire map which is displayed in the workspace. Second, the analyst can select an area and cut it in order to focus his analysis to a special part of the map which is of interest (the rest of the map is temporally removed). This function also recalibrates the shades-shades of the display, i.e. the map unit representing the greatest (smallest) dissimilarity to the documents of the considered area is set to black (white), thus enabling to better recognize structural information of detail parts (cf. chapter 6.2.3.2).

Another method is to scale the self-organizing feature map (on which the map display is based) or a user-defined part of it (cf. chapter 6.2.3). Therefore, additional units are inserted into the neural network, the weight vectors of which are defined by linear interpolation between the neighboring, originally trained weights. Once this enlarged network is visualized, the map display shows the structures on a more fine-granular level, and document points which were close together beforehand are separated to some degree. This feature is helpful for a detailed analysis of densely populated areas. Finally, a new sub-map can be generated for a selection of documents, so that only the similarity structure of a subset of texts is displayed, enabling a detailed and exclusive analysis of certain parts of interest.

7.4.2 Characterizing Document Groups: Term Profiles and Summaries

Understanding the structure presented by the document map and quickly identifying areas of interest requires some means for characterizing the text clusters. DocMINER offers three ways to do this: First, document groups can be inscribed by some relevant keywords in the map display. Second, a distribution of statistically important terms of an arbitrary document group can be computed and displayed (so-called ‘term profiles’). Closely connected to this, important sentences of groups can be extracted from the corresponding documents and displayed as a kind of descriptive ‘group summary’. In any case, the user selects documents in the map display and requests one of these group characterizations. The reason for this manual procedure (instead of using clustering algorithms) is that interpreting the structures of the map is an intellectual process which should be left to the analyst in the intended application context and thus requires flexible means of grouping anyway.

All group characterization methods proposed here are based on a term vector indexing of documents (cf. chapter 6.2.1.3). This does not undermine the exchangeability of the semantic document analysis module which is used for setting up the map since such an indexing is only computed to allow certain interaction methods. Therefore it is important to note that the group characterizations in general do not explain *why* the considered documents have been grouped by the map. However, since a basic assumption is that similar documents do share some keywords or at least contain descriptive terms which are statistically significant (no matter if the semantic document analysis component used exactly this information for determining the documents' similarity), using term vectors as the basis seems appropriate.

Thus, the basis for the different characterization functions are term profiles. Term profiles use simple measures to determine statistically important terms of a given document group. Let D denote the set of all document term vectors d of the collection, and let Γ be a (user-defined) group of documents, $\Gamma \subseteq D$. As the simplest measure, the sum profile of Γ is defined by

$$p_S(\Gamma) =_{\text{def}} \sum_{d \in \Gamma} d.$$

The k terms t_{i1}, \dots, t_{ik} with the highest corresponding weights in $p_S(\Gamma)$ are called 'significant terms' (this profile is proposed in [CKPT92] to compute a so-called 'cluster digest' for scatter/gather browsing). The sum profile indicates keywords which are relevant in the considered group according to the weighting scheme used for encoding the documents as term vectors. The relation profile of group Γ , $p_R(\Gamma)$, is defined as

$$p_R(\Gamma) =_{\text{def}} \frac{(p_S(\Gamma))^2}{p_S(D)}.$$

The k terms t_{i1}, \dots, t_{ik} with the highest corresponding weights in $p_R(\Gamma)$ are called 'characterizing terms'. These keywords are more significant in this group than in others, according to the sum profile. A drawback of the relation profile is that terms which are very significant in a few documents of the group – and appear *only* there – may dominate the profile. This effect can be lowered by multiplying the values of $p_R(\Gamma)$ by the fraction of documents in Γ which contain the corresponding term, yielding the group profile $p_G(\Gamma)$.

In DocMINER, the term profile of a user-selected group, i.e. the k terms and their weights according to a chosen profile, is presented as a bar chart (see figure 7-5). The number k of terms to be displayed can be defined flexibly by the analyst. A selected group can also be inscribed in the map display by some relevant keywords (figure 7-4). For this, the selected area of the map is subdivided into a grid structure. Each cell of the grid is regarded as a subgroup, and the l top-weighted terms of this subgroup (according to the chosen term profile) are displayed in the map. In contrast to the more comprehensive term profiles, the inscription function can be used to get a quick idea of the fields contents before interesting groups are examined in more detail.

Another group characterization realized in DocMINER extracts 'important' sentences from the documents of the considered group. This method is related to the field of automated text summarization which is concerned with producing a comprehensive overview of single or multiple documents. There are many different types of summaries, depending on the specific application area (e.g. presenting information retrieval results, information extraction, or personal information agents). Hovy and Marcu [HoMa98] distinguish 5 different dimensions of summary genres: indicative vs. informative summaries, depending on the 'depth' of presenting information, extract vs. abstract, referring to the kind of presentation (i.e. just listing fragments or re-phrasing contents), generic vs. query-oriented, i.e. contents- or query-driven

Table 7-1: Simple methods for topic extraction from texts (according to [TeMo97])

Method	Basic Idea	Procedure
word-frequency method	Important sentences contain frequently occurring words.	Increase the score for sentences containing characteristic terms.
location method	Important sentences occur at the beginning or the end of texts.	Take the first (the last) sentences as extraction.
optimum position policy	Position of important sentences depends on type of text.	Learn these positions by considering overlap between sentences and indexing terms.
cue-phrases method	Important sentences contain marking phrases, such as “this paper is about...”.	Extract sentences containing these cues, using a list of (positive and negative) indicator phrases.
title-based method	Titles indicate important topics.	Score the sentences by the relative frequency of title words and extract top-scoring sentences.

summarization, background vs. just-the-news, taking into account the state of knowledge of the user, and single-document vs. multiple-document source.

In the context of field characterizations for document maps we are interested in an indicative and contents-driven extract of information from a multiple-document source, which is primarily used for giving the analyst a quick overview of the contents of a text group. The user’s background knowledge is not considered, and thus only topical information is extracted. Interpreting the contents and generating new sentences would be far beyond the scope of this feature in the context of this work. Table 7-1 gives a brief overview of simple methods for topic extraction from literature (cf. [TeMo97] for a survey and evaluation). The method used in DocMINER is a combination of word-frequency and location method, but the other methods could be incorporated in future extensions as well. The summarization algorithm extracts important sentences as follows: For all documents of group Γ the first n sentences are parsed and a score ϕ for each sentence s is computed, where ϕ is the sum of weights of indexing terms in s according to the chosen term profile $p(\Gamma)$ of the group, normalized by the total number of indexing terms in s . Then, the k top-scoring sentences of the group are displayed. Clicking on a sentence in the interactive workspace highlights the document in the map from which it was extracted (cf. figure 7-5).

7.4.3 Coupling a Query Interface with the Map Display

The query interface – as a standard retrieval component – allows to pose keyword-based requests against the text collection and thus supports goal-directed searches. DocMINER supports four different types of queries: Boolean search, vector search and full-text search as standard methods, and a so-called ‘map query’ as a non-standard query type. Query results are not only displayed as ranked list, but there are also map-oriented visualization techniques which realize a close coupling of the query interface with the visual map workspace.

For the Boolean and the vector search, documents are encoded as term vectors $d = (d_1, \dots, d_N)$ where $N = |T|$ for the set T of indexing terms (cf. chapter 6.2.1.3). The weighting scheme and similarity functions used are defined by the parameters of the statistical indexing module. A Boolean query Q , consisting of atomic terms t and connectors AND, OR, NOT and the additional operator NEG, produces a result set of ranked documents according to the retrieval value function $rv_d(Q)$ for document d (cf. table 7-2). The additional operator NEG can be used instead of NOT in order to ‘punish’ documents which contain certain terms rather than rigorously excluding them. A vector query Q is a list of (possibly) weighted terms t , i.e. $Q = t_1(w_1) \dots t_k(w_k)$, $w_j \in \mathbb{R}^+$, e.g. ‘hubble (5) space (1)’. Such a query is regarded as a term

Table 7-2: Retrieval values of Boolean queries in DocMINER. The result set contains all documents d with $rv_d(Q) > 0$. The function $stem(t)$ yields the word stem of the query term t .

Boolean query Q		retrieval value $rv_d(Q)$ for document d
t	for a term t	d_j if $d = (d_1, \dots, d_N)$ and d_j corresponds to $stem(t) \in T$
$A \text{ AND } B$	for Boolean queries A and B	$\min(rv_d(A), rv_d(B))$
$A \text{ OR } B$	for Boolean queries A and B	$\max(rv_d(A), rv_d(B))$
$\text{NOT } A$	for a Boolean query A	$0 : \Leftrightarrow rv_d(A) > 0$
$\text{NEG } A$	for a Boolean query A	$\max \{rv_d(A) \mid \forall d'\} - rv_d(A)$

vector $q = (q_1, \dots, q_N)$ where $q_i = w_j$ if $T \ni t_i = stem(t_j)$, i.e. the word stem of the query term t_j equals the index term t_i . The retrieval value for a document d with respect to the query vector q (normalized to unit length) is determined by the vector similarity function used by the statistical indexing module. Finally, the full-text search performs a simple string matching in the original text documents. Here, documents in the result set are ranked by the matching frequency.

In all of these cases, retrieved documents are presented as a list which is sorted by their retrieval values (i.e. the assumed relevance of documents regarding the query). The retrieval values are also visualized by bar charts, helping to better assess the relevance structure of the result set. In addition to this ‘classic’ way of presenting query results, matching documents are highlighted in the map display. The analyst can define a lower bound for retrieval values using a slide control, so that only documents with retrieval values above this threshold are highlighted. This kind of visualization can help to identify relevant documents more easily, e.g. by considering clusters of highlighted objects (cf. chapter 5.3.3), allows to understand and explore the context of matching documents, and helps to identify regions in the map which deal with interesting aspects. Results of queries can be saved and the highlighting can be restored and superimposed by the highlighting of results of a second query. This visual overlay technique allows to analyze the collection regarding overlapping topics. A special visualization technique is implemented for the vector search: A so-called ‘link network’ connects all pairs of documents which match a given vector query by line segments (see figure 7-6). The color of these lines indicates the degree of similarity of the documents *regarding the query terms*. Again, a selection threshold defines a lower bound for the display of similarity links. This visual interaction method allows to superimpose the similarity structure defined by the map with a user defined similarity structure.

Finally, there is a purely map-oriented query method: The so-called ‘map query’ assigns a piece of text, e.g. a list of terms, a text fragment or even a complete document, to the point in the map display which best reflects the similarity to all documents of the collection. Therefore, the query text is indexed and compared against each document of the collection by the document analysis module (cf. section 6.2.1). Then, the spatial scaling module computes a representative for the text object in the semantic document space (cf. chapter 6.2.2), and finally the topology preserving mapping module determines the best-matching unit for the representative in the self-organizing map (cf. chapter 6.2.3). This query type can be used, e.g., to find a starting point for exploration if a certain ‘prototype’ document is given.

7.5 Usability Criteria Met by the Prototype

In sections 2.3.2.2 and 7.1 Shneiderman’s ‘Visual Information Seeking Mantra’ and the seven elementary tasks which should be supported by advanced user interfaces have been presented.

Table 7-3: Categorization of selected interface functions

type	interface functions
zoom	scale map area, standard zoom
filter	compute sub-map, cut map area
details-on-demand	compute and display important sentences, compute and display term profile, display full-text, inscribe document points with titles, inscribe field with keywords
relate	add new document, map query, color-code documents, pose query, link-network, highlight documents
extract	define sub-collection by directly selecting items or transforming query result into selection

By discussing the realization of these tasks in DocMINER, this section shows that the prototype meets important usability criteria. Table 7-3 categorizes selected interface functions according to the task types.

The first criterion to be discussed is the possibility to gain an *overview* of the entire collection. This is, of course, the central aim of the document map. The visual workspace allows to grasp the structure of the document space and provides direct access to every document of the collection. Keyword inscriptions can help to get a rough idea of the groups' contents. Alternatively, an additional list of titles can be used to open documents. Thus, a quick navigation through the collection is possible. Furthermore, as recommended by Shneiderman, the combination of visual map workspace and base interface allows to view details of selected items while the context of these items is still visible in the overview presentation. DocMINER offers two functions for *zooming* in on items of interest, namely standard zoom and scaling. Both functions allow to control the zoom focus and the zoom factor. When the zoom factor exceeds the screen size a miniature map preserves the overview of the entire map.

DocMINER allows to *filter* out uninteresting items and thereby to control the contents of the display by cutting out and displaying detail areas. More flexible, sub-maps of any extracted subset of documents can be generated and exclusively considered. *Extracting* a sub-collection of documents can be done by directly selecting items or by transforming a query result into a selection. For any selection of documents, *detail information* on various levels of granularity can be displayed in the separate register field or in the visual workspace whenever necessary. The register layout allows a quick change between different detail views.

Relating documents is realized in the prototype system in several ways. First, of course, the metaphor of the display allows to relate documents according to their similarity: The overall similarity relationship between the documents of the collection and, moreover, between the given texts and a query document can be examined (add document, map query). The possibility to color-code documents individually allows to examine relationships between the inherent document similarity structure and any kind of externally defined categorization. Highlighting documents, either automatically by a query or manually by specifying document names or categories, sets the focus on special items and allows to relate some external specification to the similarity structure. Finally, the link-network in the vector search function visually relates documents according to a list of terms.

The prototype system does not keep a full *history* of actions in order to support 'undo' operations. However, query parameters can be permanently saved and a record of queries is maintained during each analysis session in order to support progressive query refinement.

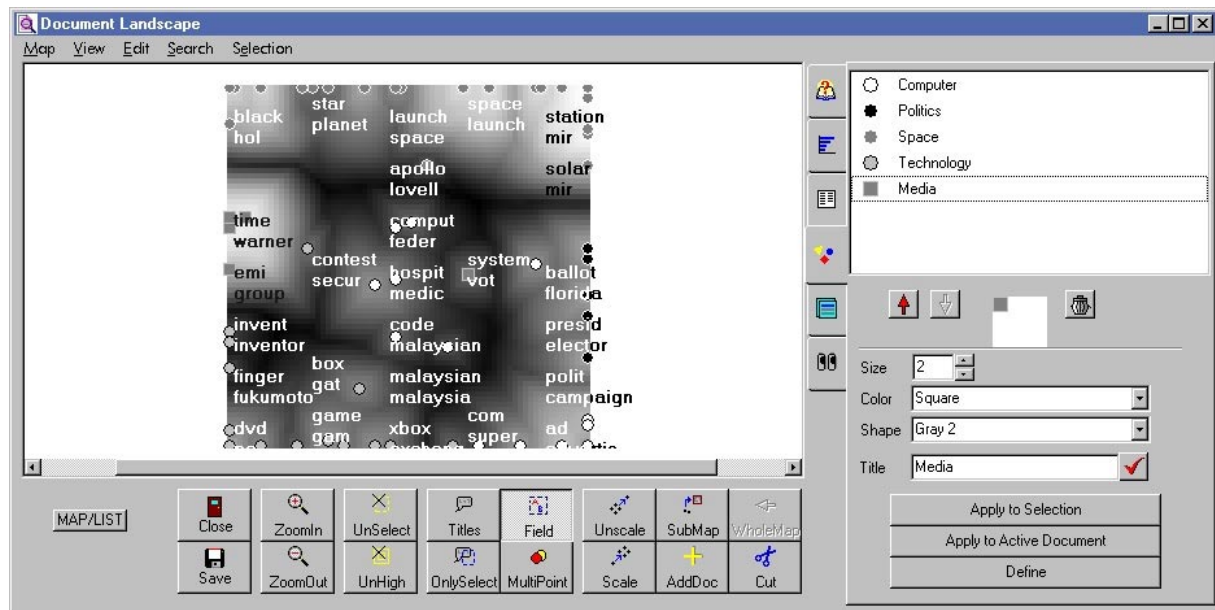


Figure 7-4: Getting a first overview of the collection: The legend register shows the defined symbols for the a priori given document classes. Document points in the map are color-coded regarding these classes. Fields in the map are inscribed by the top-scoring keywords of the underlying manually selected document group (selection frames are hidden for a better readability).

7.6 A Sample Session

This section aims at presenting the interactive features of the prototypical document map system in action in order to provide a feeling for its practical usage before the overall document map approach will be evaluated in detail in the remainder of this thesis. Therefore, assume the following simple scenario: An analyst wants to get an overview of a collection of documents, understand its inherent structure and relate it to a given categorization of texts. In this example a small collection of 64 articles from CNN online (www.cnn.com), derived from different categories, is used.

First, the analyst defines a new project in DocMINER and generates a document map. Using the hints for setting computation parameters from chapter 6.4, he decides to use the term frequency weighting scheme and the cosine measure of similarity for the document analysis module, and a self-organizing feature map with a 70×70 grid, trained with 20 training steps per document. In order to determine a good number of dimensions for the semantic document space he computes a stress curve prior to training the SOM, which shows that 34 dimensions cause a stress of less than 10% and thus are completely sufficient. Following the computation of the document map, the analyst enriches it with the given class information: The document's titles contain an identifier for each given category. In order to examine the relationships of classes later, he color-codes the documents in the map according to these categories. Therefore, he uses the interactive legend to define symbols for each class and assigns these symbols to the respective documents using the text list register (cf. section 7.4.1).

Then, he starts exploring the map: For getting a first impression of the collection's topical structure he selects visually identified groups of documents and request a field inscription (cf. section 7.4.2) to get a quick overview of the topics tackled in the different areas (figure 7-4). Among other observations, this first picture conveyed by the map shows that a relatively large group of documents is dealing with different 'space' issues, in particular rocket launches, space stations or black holes (northern part of the map, 24 texts according to the title list which the analyst displays while selecting coherent groups of documents). Other topics in-

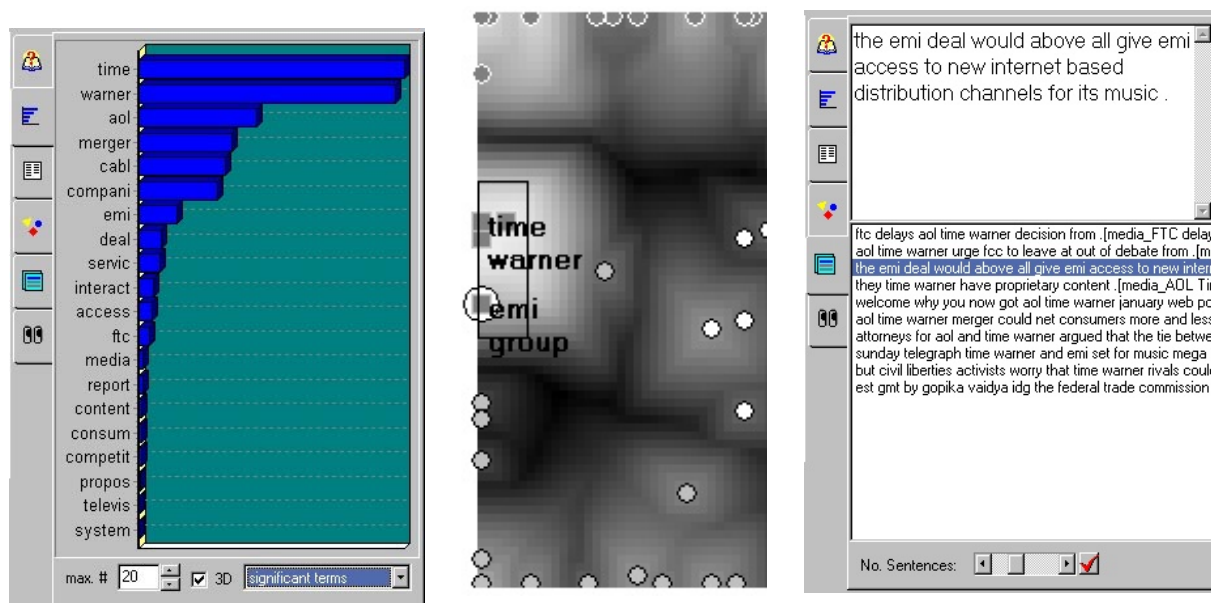


Figure 7-5: Analyzing a group in more detail: term profile of the selected group (left) and ‘important sentences’ register (right). The highlighted sentence in the list is presented in full in the window above, the document from which it has been extracted is highlighted in the map display (white circle, middle of the western edge of the map).

clude the Florida recount at the US presidential elections (eastern edge), or household electronic gadgets (southern area).

Having obtained an overall impression, the analyst now turns to a more detailed consideration of topics. He starts with the strongly related group of documents at the western edge of the map, labeled with ‘Time Warner’ and ‘EMI’ (figure 7-5). In order to get a more detailed overview he selects this group by drawing a selection frame, scans the titles of the documents in the title register and displays a term profile of the group, showing the significant terms according to the sum profile (cf. section 7.4.2). Seemingly, most documents of the group deal with the AOL Time Warner merger. In addition, the user displays important sentences extracted from the text for better understanding the concrete contents. He clicks on an interesting sentence in the list, so that the full version of the sentence is displayed and the corresponding document is highlighted in the map. He repeats this procedure for the remaining groups in order to quickly grasp the collection’s topics.

Now that the analyst is sufficiently familiar with the text corpus he starts with relating the collection’s inherent similarity structure to the given classes. He already has color-coded the document points according to the categorization of the corresponding texts and now switches to the interactive legend register again (cf. figure 7-4). By clicking on the legend entries he successively highlights the documents belonging to each class, which helps to easily recognize the distribution of corresponding texts within the map. The analyst neglects classes which are located in coherent, more or less separated areas (like the space or the politics documents), and concentrates on categories which overlap to some degree or have outliers. He identifies three documents which are not located within the group of other texts of the same class and sets markers to better find them again (figure 7-6).

The analyst opens the rightmost marked document: It is a text about the opportunity for technology vendors to bring electronic voting systems to market due to the Florida presidential voting fiasco – obviously a close relation of the document from the computer category to the group of politics texts. The second marked document (in the middle of the three) is a text from the media category within documents of the computer group. The analyst selects the text and opens the ‘important sentences’ register in order to quickly get an impression of its main

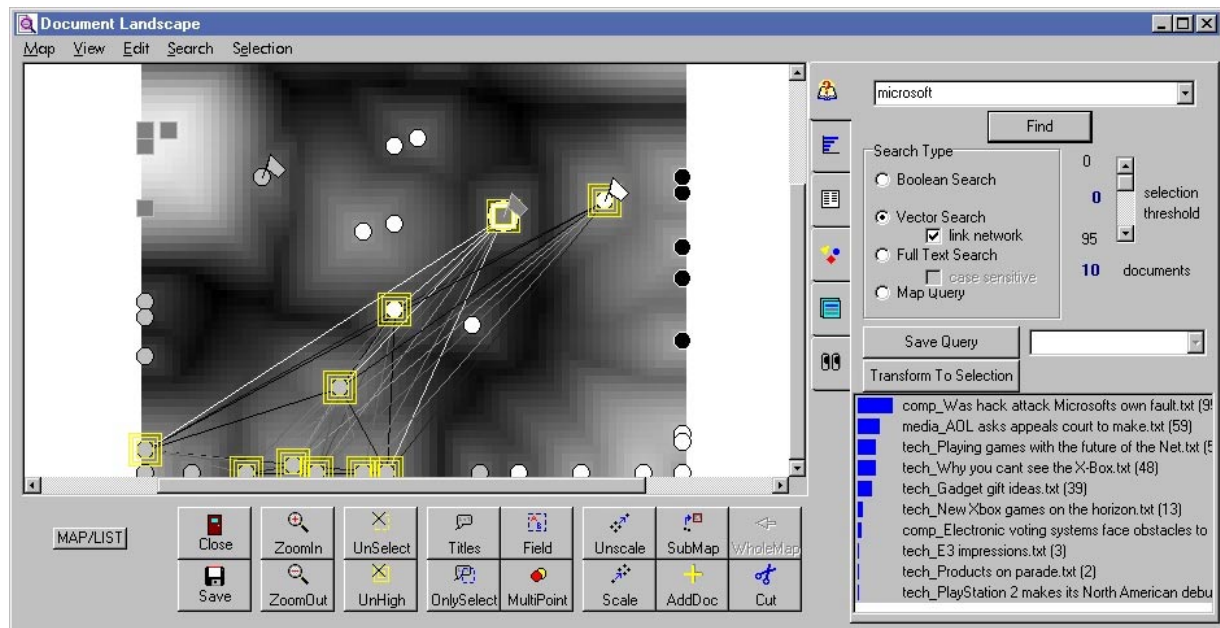


Figure 7-6: Relating the inherent similarity structure with a given categorization: Three documents are highlighted with markers. From left to right: a technology document near the media group, a media text within the computer group, and a computer document near the texts about politics. The concentric squares highlight documents that match the vector query ‘Microsoft’. A network of colored line segments indicates the relationship of texts according to the query term: The brighter the color, the more similar are the documents in this respect.

topics without having to scan the complete document. From that he derives that the text is about AOL filing a brief to support government’s actions in the antitrust case against Microsoft. He wonders which other documents might deal with Microsoft and uses the query interface. The analyst decides to pose a vector query with the simple search term ‘Microsoft’ and applies the option to draw a link network (cf. section 7.4.3), showing the degree of similarity of all matching document pairs exclusively according to the query (figure 7-6). Using this information he browses through the documents to find out what these texts are about before he turns back to the last marked document and tries to figure out its relation to the media documents in detail. This is the point where we leave our analyst alone since we have already seen most interactive features in action.

8 Case Studies

The motivation of developing the proposed document map method was to effectively support the exploration of specialized, moderately sized corporate document collections. As the experiments presented in chapter 6.5 and appendix A have shown, the proposed technology produces meaningful insights into the structure of complex text collections. Yet, an important question is whether (and how) the document map approach adds value for the analysis and exploration of documented knowledge in real-world knowledge management tasks.

This chapter contributes to a deeper understanding of the usefulness of visualizing the semantic structure of corporate text collections. The goal is to study what analysis tasks can benefit by document maps and to qualitatively evaluate to what extent the visual display can support the work. In the following, some extensive and practically oriented case studies are presented which have been performed in scientific and industrial environments. Each of the following subsections deals with a detailed case study and starts with an analysis and presentation of the application domain under examination. Besides a discussion of the application context, a method for using the document map approach in the considered problem area is proposed and an application to some sample collections is studied. Quality and benefit of the maps are examined before a case-specific conclusion is drawn.

The first of the following case studies is concerned with supporting the analysis of requirement scenarios in an international software engineering project (8.1). The focus is on exploring and structuring informal, collaboratively written use cases in order to understand the relationship of scenarios and the interaction of modules. The second case study deals with quality assurance for technical product documentation (8.2). More precisely, the use of document maps for checking the consistency of the topical structure of user manuals and for condensing documented knowledge and defining single information sources is studied. Finally, a case study in the context of a cultural research center is presented (8.3). Here, the aim is to support early phases of terminology work.

8.1 Case Study I: Structuring Use Cases in Software Engineering

In the context of software engineering, scenarios are valuable for supporting communication among system developers in the initial phases of the requirements analysis process. But the problem of how to fruitfully deal with large informal or semi-formal scenario collections consisting of weakly structured texts is still a key research issue. This section studies the use of the document map approach for automatically structuring collections of semi-formal, textual requirement scenarios in scenario management. In particular, it will be discussed how document maps can help to support the analysis and maintenance of scenario collections. To evaluate the use of document maps in this context a case study within the CAPE OPEN project has been performed. In CAPE OPEN the chemical and oil industries have defined standards for a component-based approach to process simulation. Use cases were generated in order to understand the way process simulators are used and software components interact.

Results of manual analysis work done by the project engineers and scientists are compared against results derived with the help of the document map approach. Parts of this section have been published in [BeKo99].

8.1.1 Managing Collections of Requirement Scenarios

8.1.1.1 Use Cases in Software Engineering

Understanding requirements of software systems involves a careful analysis of the domain, the scope, and the tasks of the target product. To support the complex process of requirements elicitation the use of scenarios has attracted researchers from fields like requirements engineering, human-computer interaction, or information systems.

Scenarios, in software and systems engineering, describe processes and interactions related to a system under examination. They exist in a variety of forms, e.g. as video scenes, examples of existing products, or narrative use and context descriptions. For a classification framework for different scenario approaches cf. [RBC+98]. Roughly spoken, scenarios contain situations which reasonably might occur. It is commonly accepted that developing scenarios is a valuable approach for supporting communication in the initial phases of system development and, furthermore, that scenarios are a key element of change management [WPJH98].

The use case driven scenario method by Ivar Jacobson [Jaco92] has increased the interest in scenario management especially in object-oriented software design. *Use cases* describe how an external actor uses a system to complete a certain process. More precisely, a use case is a semi-formal textual and narrative description of an action that is performed by an external actor on a specific part of a system. The actor can be a user or another module or object of the system. The interaction between system and actor and the subsequent actions are described. This description may include, for example, messages passed to the system or in- or output from or to the actor. Hence, the use case is attempting to capture the logical interactions rather than the physical appearance of the system. A *use case model* consists of a collection of inter-related use cases. The Unified Modeling Language (UML) provides a notation for modeling systems using object-oriented concepts, including a standard notation for use cases [RJB97]. Use cases are no requirements specifications by themselves but imply requirements by the stories they tell [Larm98].

8.1.1.2 Handling Collections of Use Cases

It is quite well understood that scenarios support communication between developers, but there are still a lot of important research questions regarding the complex and interdisciplinary scenario approach (cf. [JTC98]). Among them, the question of how to deal with large collections of weakly structured informal scenarios – such as use cases – is crucial.

Lessons learned from a computer-supported use case approach to the development of interoperation standards for process simulators in the chemical industries stress this observation [JBK+99]: Formal UML representations are hard to understand during their design. This holds especially, if the team of developers is widely distributed both geographically and in time. On the one hand, the more intuitive and narrative use cases proved to be the central medium of communication and agreement more than the formal representations. On the other hand, even in a seemingly limited domain the number of use cases quickly grows, with all the resulting risks of redundancy, inconsistency, and inadequate mapping to formal specifications and implementation standards.

Moreover, scenarios typically evolve over a long period of a system's life cycle and, thus, the problem of maintaining them in a repository arises. Problems coming along in the context of handling collections of use cases originate from the following facets of scenario management:

- *Collaborative aspect:* In many projects the production of scenarios is a cooperative and distributed venture. In particular in world-wide projects, such as standardization efforts, the cooperative aspect has a very special weight. In addition, the growing importance of the World Wide Web as a global market place accelerates the world-wide distributed production of software. Teams of developers working in different places of the earth meet in virtual workrooms (e.g. using tools like the BSCW, i.e. "Basic Support for Co-operative work" [BAB+97]) and share their ideas and knowledge. There are several problems coming along with this collaborative approach. Important in the use case context is that many of the usage situations described by different authors may be more or less similar and can be found in several and rather different contexts. Sometimes these scenarios contain contradictory statements. Thus, it is very important to synchronize the knowledge elicited by distributed groups and to detect those implicit and hidden relationships in order to recognize (partial) redundancy and avoid inconsistency.
- *Analysis aspect:* In practice the number of scenarios worked out often exceeds a manually manageable size. Thus, it often turns out that it is very difficult to elicit semantic relationships between single scenarios beyond an *a priori* defined structure. To better understand the interplay among system components and to effectively discover the user needs implied in the stories the use cases tell it is necessary to gain an intuitive overview of the inherent semantic structure of the use case collection. Such a structure would offer much benefit for the interrelation and condensing process of scenario management.
- *Maintenance aspect:* During a system's lifecycle the number of interesting scenarios typically grows over a long period of time. Complex situation descriptions, possibly considered from different viewpoints, are added to the initial collection of scenarios. To interrelate and integrate these cases with the existing ones an intelligent repository would be of high value which automatically correlates semantically similar scenarios.

In this case study we face the problems sketched above. The question is whether document maps can provide a means for assisting the analysis, correlation and maintenance of complex, collaboratively produced informal scenarios. A document map of use cases from the CAPE OPEN project is presented and the results of a map-aided analysis of the collection are discussed. Before that, the next section discusses approaches to handle collections of scenarios and software components.

8.1.1.3 Structuring and Retrieval of Scenarios and Software Components

The question of how to deal with collections of software or requirements descriptions has been addressed from different points of interest. Related to managing scenario collections are methods for retrieving software components. The purpose of this section is to present some related work in the field of software reuse and requirements engineering. However, a detailed discussion is beyond the scope of this section. For a survey on software reuse systems, indexing and retrieval methods see [MMM98].

Interesting in the context of this case study are methods for searching and organizing software repositories regarding the software components' descriptions. Different techniques for text description-based retrieval have been developed: Maarek et al. [MBK91] use information retrieval techniques for organizing software libraries, Girardi and Ibrahim [Gilb93] exploit the

specialized style of software descriptions and use a frame-like formalism in order to improve the effectiveness of software retrieval.

Fischer [Fisc98] presents a method for specification-based browsing in software component libraries for reuse purposes. This approach requires the software components to be indexed by formal specifications. Based on these logical formulas a theorem prover establishes relations between single components. Finally, the user can browse through a fixed navigation structure, the so-called concept lattice. However, the formal character of specification-based browsing prevents its application to informal requirement documents.

Organizing informal representations has also been addressed in requirements engineering: Pohl and Haumer [PoHa95] propose a hypertext model for structuring informal requirement representations (e.g. multimedia documents). This formal hypertext model relates specified requirements with their sources and thus provides a tool for keeping the whole requirements engineering process traceable during a long-termed process of decision and change management. However, it is not intended to support especially the analysis task in the initial phase of projects where informal requirement documents are typically designed in a brainstorming fashion.

A document map approach for structuring software libraries according to the software object's descriptions has been introduced in [MTK94, Merkl95a]. This work has already been discussed in chapter 5.5.2. To repeat, besides other shortcomings, this type of document map does not provide much insight in the relation of documents and groups of documents.

To conclude, the problem of automatically structuring large informal scenario collections for supporting their analysis and maintenance is still an important research area. The next section considers the usage of our document map approach in the context of managing use case collections.

8.1.1.4 The Proposed Application of Document Maps for Use Case Collections

Reconsidering both, the different facets of scenario management as discussed in section 8.1.1.2 and the features of the proposed document map approach, a visualization of the semantic structure of use case collections could support the following tasks:

- *Synchronizing knowledge elicited in distributed groups.* Due to the collaborative aspect of scenario management an important step in the analysis of scenarios is to detect inconsistencies and redundancies among the collection. A use case map would help to identify similarities among different scenarios. By inspecting close relationships which are unexpected by the engineers (partial) redundancy and inconsistency may be detected much easier than by a manual analysis of the use case collection. However, once such unexpected relationships have been found the conclusions derived from that have to be drawn by the engineers. Thus, for this kind of task the map offers limited but useful assistance. It does not provide reasoning capabilities but provides an infrastructure for a structured search in the “space of documents”. The lack of more formal assistance results from the (explicitly accepted) lack of scenario formalization in early phases of system development.

An (abstract) example may clarify this consideration: A specific situation, e.g. the usage of a certain system component, may be described by more than one author by mistake. The double description may contain contradictory statements due to the fact that the respective authors may have made different assumptions regarding the context of the scenario sketched. Another situation may be, for example, that different scenarios contain similar or identical sub-cases. Again, the respective documents will tend to be located

near each other. In such a situation the sub-case could be sourced out. Vice versa, highly similar situations could be combined if desired. To sum up, by relating the corresponding scenarios the map gives a hint on where to look for critical descriptions and thus aids the synchronization of knowledge elicited by collaborative working groups.

- *Analyzing relationships of single scenarios and scenario groups.* The support offered by document maps for this aspect is closely related to that from above. In addition, the map provides a refined basis for discussion as it structures available use cases. With the graphically visualized structure at hand it is possible to detect and discuss relationships of single use cases towards each other, to define groups of related scenarios – whether directly using the cluster structure or additionally considering other rationales – or to understand relationships between just being defined or *a priori* given groups.
- *Maintaining collections of use cases.* A ‘use case landscape’ could serve as an intuitive retrieval interface for a repository of scenarios. Semantically similar cases can be found close together and, thus, the user can browse across groups of related scenarios, exploring the collection and searching for documents.

To sum up, a semantic map of use cases is expected to support the collaborative, analysis and maintenance aspect of scenario management. The priority application of document maps for textual requirement scenarios is certainly the assistance for analyzing large collections of scenarios. In the remainder of this paragraph the suitability of the document map approach for the analysis task will be examined in the context of a real-world usage scenario.

8.1.2 Use Cases for Standardizing Open Simulation Environments

In the next section the application of the document map approach for software engineering projects will be discussed. We use the use cases generated during the CAPE OPEN project as a real-world example to study the benefit of this approach. CAPE OPEN stands for “Computer-Aided Process Engineering Open Simulation Environment” and is an EU funded project with participants from the chemical industry (BASF, Bayer, BP, DuPont, ELF, ICI), software vendors (AspenTech, HyproTech, QuantiSci) and universities (Imperial College London, INPT Toulouse, RWTH Aachen) under co-ordination of the French process licensing company IFP. It aims at defining a new standard for high-performance process simulation software.

Process simulators are highly sophisticated pieces of software, designed for creating mathematical models of manufacturing facilities for processing and/or transforming materials (chemical, oil, food). These tools have become vitally important for chemical engineering companies for several reasons: The market is rapidly growing while innovation cycles are shrinking. This pressure is strengthened by a growing sensitivity for environmental issues.

The process simulators that are currently in use are closed monolithic applications. They are quite inflexible when it comes to integrating new components. Another drawback of this situation is that it is almost impossible to combine modules from different vendors into one single simulator. In practice, such a combination is of high interest due to the limitations of individual products. Therefore, the CAPE OPEN standard aims at creating a framework for open component based simulation software. To identify the components in such an open simulator a use case approach was applied [JBK+99].

Before going into detail, some background on the subsystems of a process simulator are provided and the collection of use cases as generated in CAPE OPEN is sketched. This will render a basis for the discussion of the use case structuring.

8.1.2.1 A Conceptual View on Process Simulators

Simulators differ widely in architecture and implementation but all have common functionality imposed by the underlying modeling tasks which they address. This functionality can be summarized in terms of four key ‘conceptual’ component types [JBK+99]:

- *Simulator executive:* This component is the simulator’s core as it controls the set-up and execution of a simulation. It is responsible for installing other components, registering them in a repository, managing interactions with users, accessing and storing data, and, finally, for reporting and analyzing simulation calculations. Furthermore, it is responsible for a consistent flowsheet set-up, error checking and preparatory work on solving it (graph analysis).
- *Unit Operation Modules:* These components represent the behavior of physical process steps (e.g. a mixer or a reactor). They are linked to the simulation flowsheet which represents an abstraction of the plant structure. They compute the quality of a material stream of their outlet if the according information is given at the inlet. The simulation models are assembled from predefined libraries of unit operation modules into a flowsheet which represents the overall plant and is handled by the simulator executive.
- *Physical properties packages:* An important functionality of a process simulator is its ability to calculate thermodynamic and physical properties (e.g. density or viscosity) of materials. Thermodynamic packages are complex and highly optimized pieces of software. Since they provide the basic calculations for all unit operations, the overall performance and quality of a simulator strongly depends on its thermodynamic package. It is estimated that up to 75% of the simulation time is spent for these calculations [BrJo95].
- *Numerical solvers:* This includes both the specialized mathematical methods used to evaluate the equations that describe a *unit operation* (unit solving) and the methods used to evaluate the overall flowsheet (flowsheet solving).

8.1.2.2 Use Cases in CAPE OPEN

To obtain a formal interface specification use cases have been applied as an important first step in the direction of standardizing simulator components. There are various reasons for this decision [JBK+99]: First, the use case approach is specific enough to capture the major functionality of a process simulator which has to be mapped to a software component. On the other hand, it is independent from the technology used for realizing the component model. Second, applying use cases is a powerful way to split up the functionality of existing monolithic simulation software into manageable components. Third, as mentioned in section 8.1.1.1, the use case driven scenario approach has proven to yield a basis for communication among groups of distributed developers.

In CAPE OPEN use cases are intended to describe the application of simulator components. In particular, they point out the usage of each component and define the order of running physical processing operations which represent units of plants. The three most important subsystems of a process simulator are the Unit Operations, Physical Properties and Numerical solvers. Because it is one goal of the new standard to reduce the size of the executive no use cases were created for it. Nevertheless, an important feature of a simulator executive had to be taken care of: graph analysis. A graph analysis tool (GAT) checks how a flowsheet can be solved by determining how to break up cycles in it. This coarse structure of a process simulator was the starting point for the generation of the use cases which, of course, should yield a finer subdivision of a simulators functionality. The use case model resulting from this project

phase not only implies the core requirements but also provides a basis for testing a proposed design.

The use cases used in this case study are contained in the Open Interface Specification documents of the CAPE OPEN project. The latest version of these documents is available from the Global CAPE OPEN Web Page [GCO99]. Please note that for the study earlier versions of the specification documents (as of November/December 1997) have been used. Two sample use cases are presented in figure 8-1.

Altogether, more than 160 use cases have been developed in CAPE OPEN by intensive research studies from the different major chemical industry partners in a distributed effort. Hence, there was the permanent danger of producing redundancies and inconsistencies. To overcome this problem, all use cases were agreed upon by the consortium and finally put into the handcrafted hierarchy as shown in figure 8-2. Table 8-1 gives the number of use cases for each main group. In the next section we will evaluate an automatically generated semantic map and its application for exploring the semantic structure of the CAPE OPEN use case collection. In detail, we will

SET PSEUDO-COMPONENT PARAMETERS FROM DATA	SET VALUE OF PUBLIC VARIABLE
<p>Principle Actor: Simulator End User</p> <p>Description: The Simulator End User requests that pseudo-component parameters be set from data. The Simulator Executive acknowledges the request and provides a facility for the user to select the streams for which data should be set in this way. If the data have already been assigned to the relevant Materials Template, the parameters are copied to the stream. If the data have not been assigned to the relevant Materials Template, the pseudo-component properties routine is requested to fetch the data. Relate physical properties data and pseudo-component parameter set to copy the data from the relevant physical properties data set.</p> <p>Exceptions:</p> <ul style="list-style-type: none"> • No pseudo-component routine has been provided. • No data have been assigned <p>Note: The implementation of this requirement would be simulator dependent. Some simulators may automatically initialize pseudo-component parameters in all streams without any explicit user action.</p>	<p>Principle Actor: Simulator Executive</p> <p>Description: The simulator executive passes a variable value and identification to the flowsheet unit, and asks the unit to update the value of its corresponding public variable with the value passed. If the unit recognises the identification, it attempts to update the recognised variable with the value passed. If it cannot update the variable (e.g. because the value passed is of the wrong type) it informs the simulator executive of this. If it does not recognise the identification it informs the simulator executive that the identification was not recognised.</p> <p>Note: This use case does not make use of information ports, and it therefore provides an alternative means of accessing a units public variables.</p> <p>Exceptions:</p> <ul style="list-style-type: none"> • Identification not recognised or is incomplete/bad. • Cannot update (value of wrong type, value out of range, or variable is read only....etc.)

Figure 8-1: Sample use cases from the CAPE OPEN project

Table 8-1: Number of use cases per main group

Use Case Group	# Use Cases
Physical Properties	74 (44%)
Unit	21 (12,5%)
Solver	45 (27%)
GAT	28 (16,5%)

- assess the computed structure by discussing the map contents on a general level,
- use the map to explore the collection, and finally
- compare and refine the hand crafted grouping with the structure visible in the map.

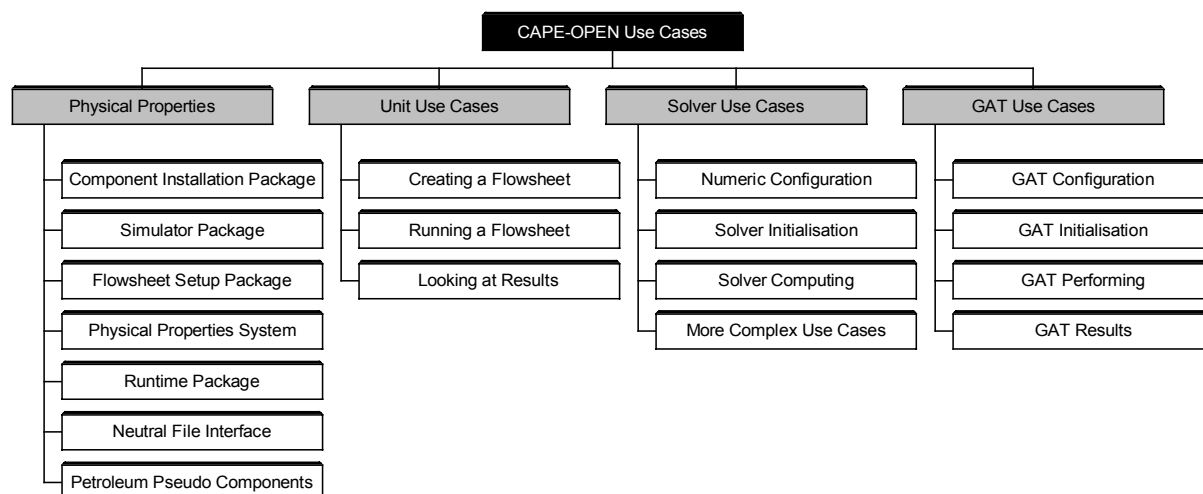
The analysis presented in this work was supported by a research assistant engaged in the CAPE OPEN project and the succeeding Global CAPE OPEN effort.

8.1.3 A Semantic Map of Use Cases in CAPE OPEN

To generate the semantic map of use cases the textual use case descriptions were extracted as pure ASCII text from the Open Interface Specification documents. Each use case forms a single document. The standard statistical indexing module was used to compare the texts. In addition to the standard list of common words, domain-specific stop words (like CAPE OPEN, port, input, definition, etc.) and key words for the use case slots (e.g. principle actor, description, assumptions, ...) have been added to the stop word list. Table 8-2 summarizes the parameters used for structuring and gives an overview of important statistical facts of the collection as well as the map.

8.1.3.1 An Overall View of the Use Case Map: Evaluation of the Structure

In this section we discuss the CAPE OPEN use case map on a general level before performing a more detailed analysis of its content. Here, the driving question is whether the global structures presented by the map are semantically sound and thus could yield a reasonable basis for

**Figure 8-2: Hand-crafted use case hierarchy for CAPE OPEN**

the tasks sketched in section 8.1.1.4.

Looking at the ‘use case landscape’ in figure 8-3 (a) the user can identify five major areas (I–V), each of which is subdivided into smaller areas containing sub-groups of documents. Figure 8-3 (b) presents the same map of use cases, but this time the document representatives are marked by an icon which identifies the (given) main group the respective use case belongs to (cf. figure 8-4). The areas in the map are separated by ‘deep’ dark ditches, roughly corresponding to the *a priori* subdivision.

Taking a closer look we can identify the physical properties use cases in the south-eastern area of the map. The numerical solver use cases are located in the north-eastern part and the center, the GAT use cases can be found in the south-western sector. Finally, the unit operation use cases are on the northern edge of the map. Apart from this general and coarse view there are some areas where use cases of different type are mixed.

First, we look at the north-western corner as well as the lower eastern edge of the map where a mixture of properties package and unit operation use cases can be found. As mentioned in section 8.1.2.1, the routines offered by a physical properties package form the basis of the calculations performed in a unit. The mixture of both use case types reflects this fact. In particular, these use cases describe on the one hand that a unit calls a calculation routine (e.g. calculate pressure, temperature,...) within a properties package. The according properties package’s use cases describe a function call from a unit.

Now consider the north-eastern corner of the map (cf. figure 8-5). Here, unit use cases and numerical solver use cases are mixed up. The reason is that unit operations do not only perform pure thermo dynamical calculations carried out by a properties package but sometimes have to solve difficult equation systems. This is the connection between both use case types: The unit calls the numerical solver to create an equation system, chooses some initial values

Table 8-2: Parameters and statistics of collection and map

collection	name	CAPE-OPEN Use Cases
	# documents	168
	avg. # words	139
indexing	size of vocabulary	825
	avg. # indexing terms / document	28
	weighting scheme	TF
	similarity measure	cosine (metric)
doc. space	dimension	40
	avg. stress	0.04
SOM	size of SOM	100 × 100
	training steps per document	30
	total computation time	5:48 min

and starts the solving process. On the other hand the solver accepts these calls and returns the values of its calculations. These facts are reflected by the conglomerate of different use cases in the north-eastern corner.

Finally, the graph analysis tool (GAT) use cases (south-western area) are strictly separated from the other groups. The GAT is a global pre-processor tool which defines the flow of material and energy and is used to prepare the actual flowsheet solving. In its functionality it is merely independent from the remaining modules. Thus, this separation is reasonable and semantically sound.

To conclude, the map expresses a semantically plausible interrelation between the *a priori* defined groups of use cases and thereby yields additional information on the relation of modules towards each other. This fact allows the user to derive that the four main groups are not standalone but are somehow connected.

8.1.3.2 Using the Use Case Map to Explore the Structure of the Collection

Now we sketch how the document map can be used to derive a refined hierarchy of use cases

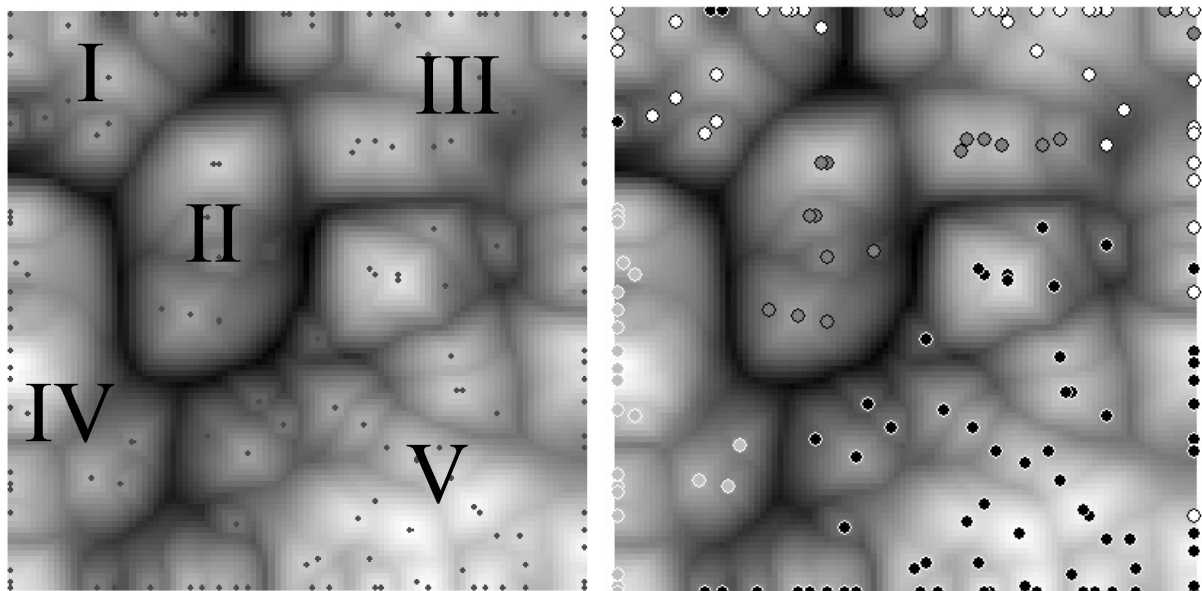


Figure 8-3: (a) Map of CAPE OPEN use cases, (b) document icons marked regarding main group membership

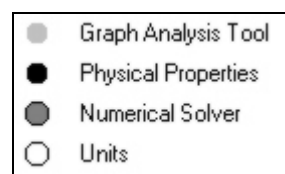


Figure 8-4: Coding-coding of use case groups

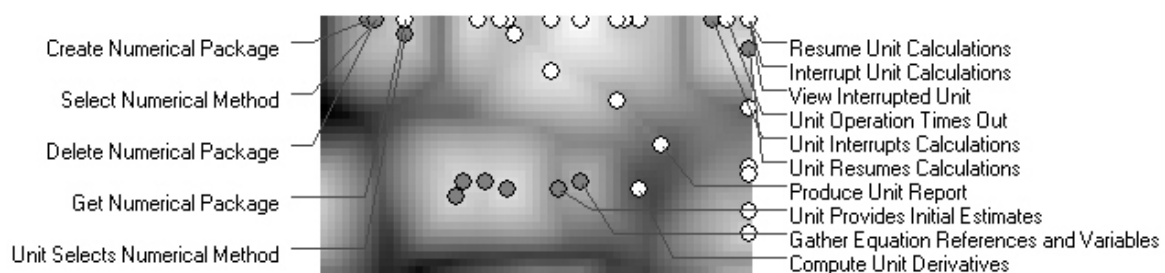


Figure 8-5: North-eastern map corner: numerical solver and unit use cases

from the automatically computed structure. Therefore, an exemplary examination of two given use case groups will be performed. To aid the process of defining groups we use the document map as an orientation guide rather than simply adapting the suggested structure. The reason is that a use case grouping should reflect not only the similarity of documents but has to take into account additional domain and application depending aspects.

Figure 8-6 shows the GAT use case part of the document map. The document icons are inscribed by the titles of the corresponding documents. Visually, five groups of GAT use cases can be identified (1–5). The inspection and interpretation of the document subgroups can be aided by considering the term profiles of each group (as derived by the DocMINER tool, cf. section 7.4.2). Table 8-3 summarizes these keywords and gives a possible interpretation for each group. Accordingly, the group of GAT use cases could reasonably be divided into 5 subgroups, namely *GAT control* (1), *GAT configuration* (2), *GAT performing* (3), *GAT results* (4), and *GAT initialization* (5).

Now consider the unit use case group. Starting in the north-western corner (figure 8-7) we find a separated section of unit scenarios. These use cases describe (a) how units – more precisely: the units’ ports, i.e. their in- and outputs – are connected with each other and (b) sketch the configuration of units via public variables (which describe, e.g., the size of a tank). As a whole, this group defines the supply of units with as well as the retrieval of data and could be

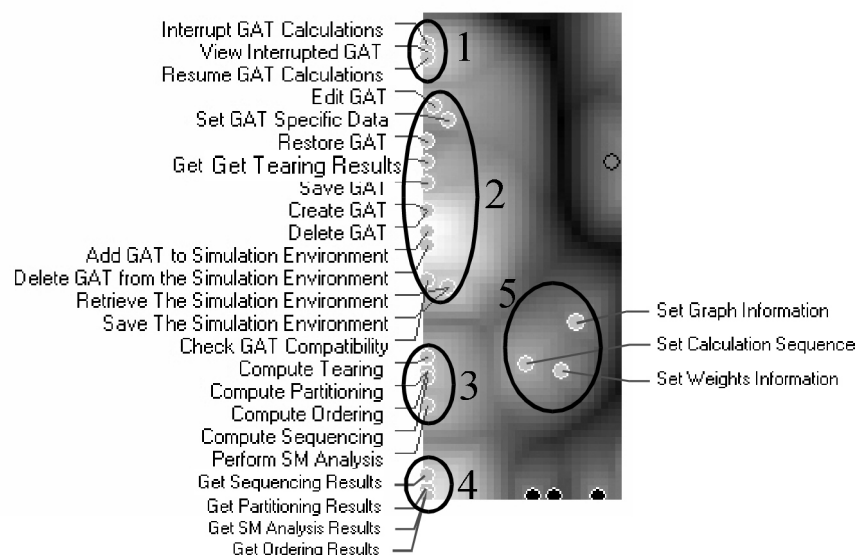


Figure 8-6: South-western area: GAT use cases

Table 8-3: Interpretation of visually identified GAT sub-groups

Group Number	1	2	3	4	5
Significant Keywords	flowsheet, calculation, interrupt, builder, resume, state	flowsheet, builder, simulation, manager, data, environment, instance, specify, save	executive, simulator, partition, perform, graph, sequence, calculation, compute, tearing, order	executive, result, simulation, compute, record, success, order, partition	simulator, executive, data, graph, weight, set, calculate, provide, record
Interpretation	control of GAT calculations	embedding GAT into the simulator's context	performing GAT calculations	retrieving GAT calculation results	Setup of the GAT

termed *Unit Data Connection*. Turning to the north-eastern part of the map (figure 8-8) the following grouping can be derived:

- Group 1 mainly describes the integration of units into the flowsheet, i.e. adding and removing instances of units to the flowsheet, managing the units' ports, or the data input into a unit which defines the unit's behavior within the flowsheet. Consequently, this group could be labeled *Unit Integration*.
- Group 2 contains two use cases concerned with defining and producing unit reports which include information about important data of the unit operation module. The third use case ("Compute Unit Derivatives") is merely about flowsheet calculations and should be added to an appropriate group. The two use cases "Define Unit Report" and "Produce Unit Report" could be added to a group named *Unit Reports*.
- The use cases contained in section 3 are obviously concerned with controlling and running calculations. An appropriate group could be labeled *Flowsheet calculation*.
- Finally, group 4 contains scenarios which describe unit validations and configuration prior to calculation (e.g. validity of the flowsheet, availability of required methods and data). A corresponding group could be named *Flowsheet Validation*.

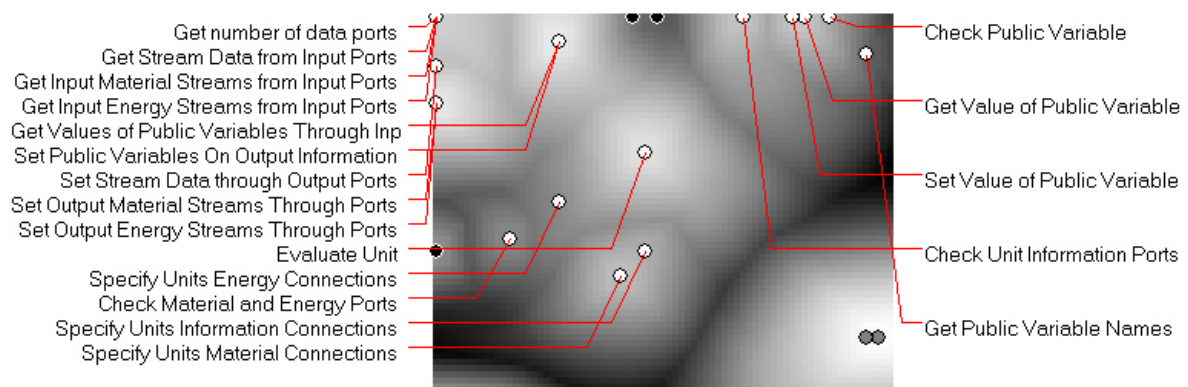


Figure 8-7: North-western corner of the map: unit use cases

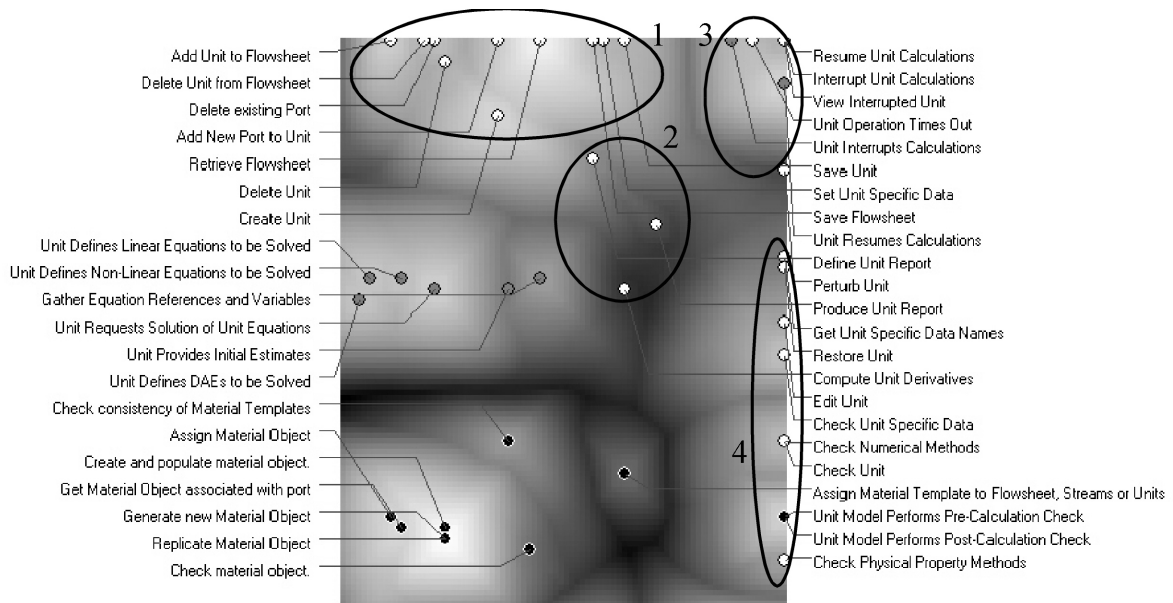


Figure 8-8: North-eastern corner of the map: unit use cases

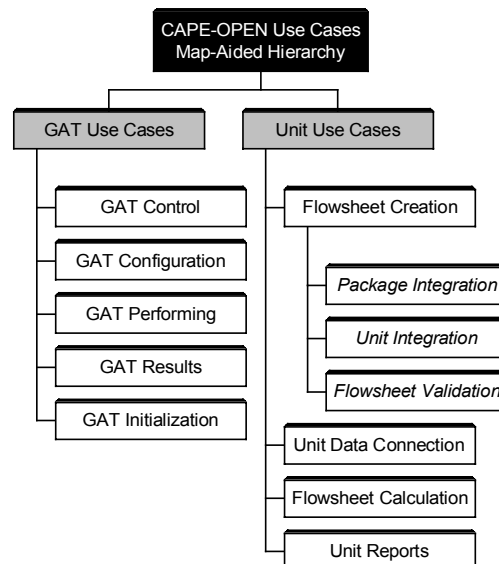


Figure 8-9: Derived hierarchy of GAT and unit use cases

Two further unit use cases can be found on the map in figure 8-3 (b): one in the south-eastern corner within the physical properties documents (“Get Physical Properties Package”), another in the northern part mixed up with solver scenarios (“Get Numerical Package”). These two use cases describe how a flowsheet is instantiated with a numerical or thermodynamic package, respectively. This explains their position within the corresponding groups. However, since the flowsheet needs the packages for its calculation, the use cases could be combined in a group named *Package Integration*. This group, along with *Unit Integration* and *Flowsheet Validation*, is necessary for creating a flowsheet. Consequently, a more general group *Flowsheet Creation* could be defined. Figure 8-9 summarizes the derived hierarchy of GAT and unit use cases as discussed in this section.

8.1.3.3 Comparing Expert Grouping and Structure Presented by the Use Case Map

Now we compare the structure visually presented by the document map against the expert grouping. Figure 8-10 presents the expert hierarchy from section 8.1.2.2 once more. This

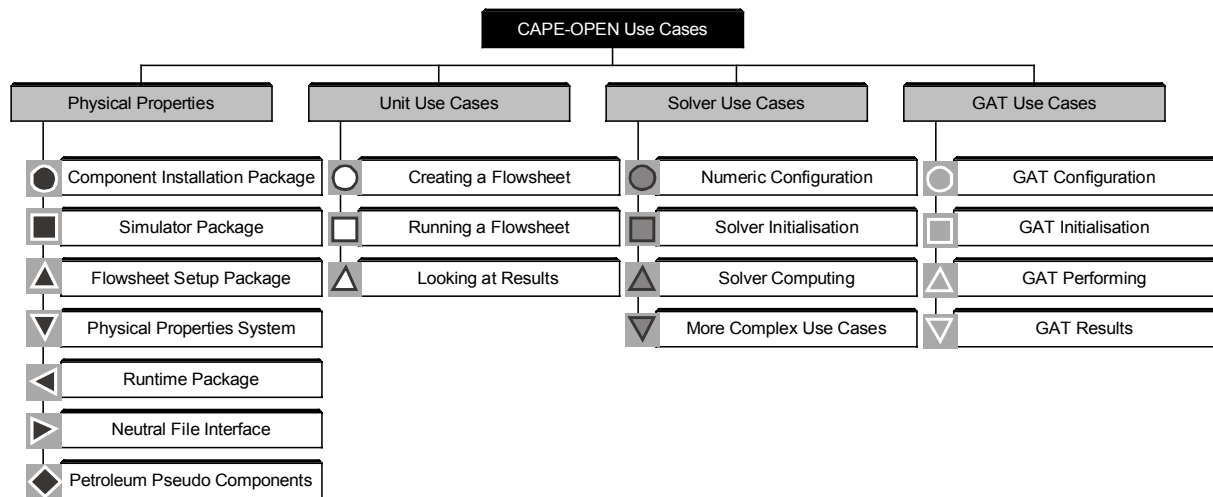


Figure 8-10: Color-coding and shape definition of use case sub-groups

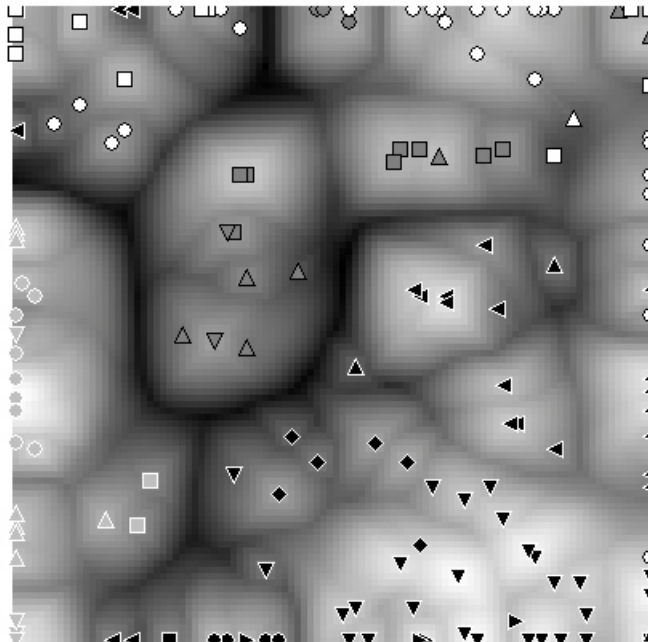


Figure 8-11: Expert grouping and map grouping

time, however, a symbol for each sub-group is given in addition. In figure 8-11 the use case map is presented with the document icons being marked regarding the membership of the corresponding document in the expert grouping.

Obviously, there is a significant correspondence between expert structure and the visual clustering. However, there are also many cases in which the visually identifiable structure is not in accordance with the expert grouping. What is the reason for this? The discussion above has shown that the document map presents a reasonable structuring of the scenario collection. The grouping derived from the map is merely *object-oriented*, i.e. based on the behavior of objects and the communication between objects, e.g. units, ports, or connections. An explanation for this grouping is that the map was computed using a document analysis model which compares texts based on keyword co-occurrence. But these keywords just describe the actions of objects involved (e.g. flowsheet builder/unit, deletes/selects/informs).

In contrast, the expert structure reflects the *sequence of actions*, namely configuration and creation, performing and calculation, and getting the results (with the exception of the rather

object-oriented *Physical Properties* group). Following this view, the use cases should be grouped regarding their function in the sequence of events. For example, consider the *Creating a Flowsheet* group which tackles how a unit is created, deleted and initialized and how it is combined with other units to model a process. The creation and deletion use cases are “Save”, “Restore”, “Create”, and “Delete Unit”. The initialization is handled in “Set Unit Specific Data”. The combination with other units is done via the ports and via the flowsheet. The corresponding use cases are “Delete Unit From Flowsheet”, “Delete Existing Port” and other use cases dealing with ports and flowsheets. In the map, the use cases of this group are clustered regarding their functionality and related to corresponding objects and actions of other groups.

8.1.3.4 Refining the Expert Structure with the Document Map

The aim of this section is to use the map in order to refine the given expert grouping. Of course, apart from the view offered by the document map, there have been design decisions and conceptual choices which were not taken into account in the document map driven exploration of the collection as presented in section 8.1.3.2. But, considering the hierarchy created by the experts, can the map offer any possible starting points for a refinement of the given structure? For the following discussion we refer to figure 8-11.

Consider the *Runtime Package* group. There is a clearly noticeable border which separates the use cases of this group in the central / eastern part of the map. Whilst the more isolated sector contains scenarios describing so-called material objects that contain the data of the flowsheet, the other use cases sketch the actual thermo dynamical calculations. Though semantically both groups belong together, a sub-grouping would be possible.

Inspecting the *Running a Flowsheet* sub-group of the *Unit* use cases we can make an interesting observation: There are two strictly separated clusters of use cases in the north-western and the north-eastern corner of the map. The eastern use cases are mixed with the solver use cases for reasons we have explained in section 8.1.3.1. These unit use cases are mainly concerned with creating and initializing equation systems that are needed *internally* in the unit. The western unit use cases describe how data from the results of these internal calculations are communicated to other units. This information is needed to solve the overall flowsheet, i.e. running the simulation on flowsheet level. This also explains that these unit use cases are located nearby the GAT use cases which form the basis of the overall flowsheet solving. Hence, we have found an additional subdivision of a manually generated sub-group which we could call *Internal Unit Solving* and *Flowsheet Solving*.

8.1.4 Summary and Conclusion

Scenarios are of high value for supporting communication among system developers in the initial phases of requirements engineering. Furthermore, they are seen as key element for change management. A crucial point in this context is handling large collections of weakly structured textual scenarios, such as use cases. In this section we have reported on the application and evaluation of the document map approach for managing the knowledge contained in scenario collections in a real-world standardization effort. The discussion has shown that the map reveals semantically correct and useful information. To sum up the major findings:

- The evaluation of the overall map structure has qualitatively shown that the map reflects reasonable relationships of the use case documents. More precisely, the plausibility of the computed structure was verified by comparing the semantics of the textual scenarios with the information extractable from the document map.

- It turned out that the map is a suitable basis for exploring a collection of informal requirement scenarios. Some analysis steps have been performed in order to reasonably refine an *a priori* given rough structure. Thus, the document map approach provides a means for assisting the analysis and correlation of complex, collaboratively produced informal scenarios. As the analysis aspect is a central problem for handling use case collections, document maps would offer benefits in this context.
- The map quite faithfully reflects the hand-crafted expert structure. However, there are also significant deviations which arise from a rather sequence-oriented view of the software engineers on the collection. Understanding the sequence of actions clearly requires additional knowledge about the domain.

Reconsidering the three aspects of scenario management (cf. section 8.1.1.2), this case study clearly points out that document maps can support the analysis process of scenario collections. Coming along with this ability is a limited support for synchronizing knowledge elicited by distributed groups of engineers (as discussed in 8.1.1.4): The process of detecting inconsistencies and redundancies is aided by the map because it structures component descriptions regarding their functionality. Thus, a focused analysis of groups concerning inconsistent assumptions would be possible. However, figuring out if these groups contain contradictory statements and inconsistencies is still a work to be manually done by human experts. Such a process could be supported by developing suitable inference techniques which would require a translation of natural language use cases into a more formal representation and would need to be supported by domain semantics. Since the automatically derived structuring has proven to be semantically sound in essence, the usage of the document map as an interface to a growing repository of scenarios seems to be feasible. To conclude, the value added by the use case map is that system engineers can gain additional insight in scenario interrelations.

Motivated by these very promising results an important and still open research question is whether the document map approach can successfully be applied in the practical work and daily routine of software engineers. Since this case study has been a “lab-experiment” without user participation – which primarily aimed at studying the principal quality and suitability of the map paradigm for managing textual requirement scenarios – a field study would be of high value which collects experiences of “document maps in action”.

8.2 Case Study II: Managing Product Documentation of Software Vendors

High-quality technical product documentation is a valuable asset of knowledge-intensive companies. On the one hand it “preserves” and makes explicit considerable aspects of the company’s knowledge. On the other hand, complete, consistent, plausible and readable documentation creates value in different areas of a product’s distribution: It can help reducing costs, e.g. for translating the documentation into several target languages or for updating the documentation when technical changes occur. Some experts regard high-quality end user documentation as a non-neglectable sales factor. Furthermore, legal aspects, such as product liability, require an accurate and consistent user documentation.

This case study reports on the application of the document map approach for managing user documentation of technical products. It has been performed in cooperation with technical writers and a consulting agency for software vendors. In section 8.2.1 some scenarios for the application of document maps for managing product documentation are discussed. These scenarios have been elicited from interviews with technical writers, consultants and practitioners

and a series of workshops of the special interest group “Technical Documentation” of Aachen’s regional industry club REGINA, held in 1998/99. Sections 8.2.2 and 8.2.3 present the exemplary application of the map approach to some collections of original product documentation and show how document maps can aid aspects of quality assurance and knowledge management. Parts of this work have been published in [BeHo99, BeHo00].

8.2.1 Managing Product Documentation – Some Scenarios

An obvious trend in technical documentation is to have user documentation available electronically, which allows quick and easy distribution in different formats and on various media. In addition, electronic documentation is intended to support enterprise publishing. The motivation is to provide access to important enterprise documents at any time and any place for every employee (cf. [Brae99]). A common basis for generating high-quality documentation of complex technical products – whether external user or internal enterprise documentation – is to define so-called *single sources*. A collection of single sources is an extensive, consistent set of information without redundancy. The information contained in these sources would then be used in multiple languages for both, printed documentation as well as electronic resources, such as online help, CD-ROM documents, Internet-based product support, or documentation used by service hot-lines. Addressees of the material may include end users as well as internal engineering personnel.

However, in practice the principle of defining single sources raises some problems. Just to give one illustrative example resulting from interviews with technical writers: In a market-leading company which manufactures machines for woodworking an individual technical documentation has to be produced for each order. Typically, different authors are responsible for the documentation belonging to the various orders. As a result many semantically similar or even identical documents are developed over the years. To achieve a consistent management of knowledge sources it would be necessary to first condense all these documents and identify related and varying sources.

To better understand the requirements for managing and defining collections of technical information sources consider the following aspects of quality assurance and information design for single sources (cf. [Amma99]):

- Different groups of users need different levels of detail. However, significant parts of the respective documentation will be relevant for more than one group. It is important to find an adequate *level of granularity* for the information pieces to be defined. In an optimal case it should be possible to combine different “atomic” information elements to a more detailed knowledge source.
- From a stylistic point of view, it is crucial not to confuse the reader by providing similar or even contradictory information. A basic principle of good documentation is to assure uniformity in style and size which allows the reader to receive the intended messages safely and quickly. This leads to the requirement of defining *homogeneous information units*.
- In both cases it is essential to *avoid redundancy* which would result in cost explosions and sources of inconsistency and confusion.
- Since the idea of creating single sources is to use them in different contexts, the editors have to *allow reusability of information units and the ability to integrate them into different context*. Thus, small and coherent pieces of information have to be defined which ideally result in a ‘unit construction system’.

In practice different scenarios can be found which are concerned with managing product documentation and supporting the creation of single sources. In the following some important examples are sketched which could benefit from an intuitive structured overview of the collection under examination.

- *Generating Technical Product Documentation:* The process of writing technical product documentation usually starts with collecting all relevant material available in a company, e.g. specification documents, informal notes of the developers, program and interface documentation, texts providing background knowledge of the application area, etc. Given such a collection, the technical writers will have to divide the documents – which usually have different degrees of granularity – into small, meaningful pieces of information. In practice this results in several hundreds of “information items” which then have to be structured and condensed to serve as a basis for the documentation to be developed. A related task would be to renew an existing documentation. In this case an author has to check the relevance and semantic context of the old texts with respect to the new documentation. Especially when dealing with older, poorly structured document collections this is usually a difficult task [BeHo99].
- *Condensing Documented Knowledge:* Technical product documentation, specification and requirement definitions and related documents contain important information about the company’s products or about procedural knowledge. Though document management systems (DMS) support the structured filing of those documents, complex relationships beyond the formal structure often remain unclear. Under pressure of time in practice important documents are often indexed poorly or filed and categorized wrongly. This leads to redundant or inconsistent documentation, and users have serious difficulties in finding the right information at the right time. Thus, it is necessary to clean up and check the valuable documented knowledge from time to time. This also includes the definition and maintenance of single sources. An important aid would be, again, to gain an overview of the material and its relationships.
- *Checking the Quality of User Documentation* regarding completeness, redundancy, or semantically correct positioning of text fragments, in practice requires an expensive effort of manual reading. Quality and readability of the documentation depends on a clearly recognizable thread: Can the reader figure out the semantic correlation between single sections? Are there any inconsistencies or redundancies in the descriptions? Is the documentation complete, or are important aspects missing? A requirement for a readable documentation is that single passages must not be semantically isolated, i.e. each section has to have a relation to its preceding sections. Getting an intuitive overview of the documentation’s semantic structure beyond the table of contents could be a means to effectively assist this analysis process. The document map approach could help here: Successive passages should be found in common areas of the map. Of course the map does not indicate whether the text is “readable” or not, but could give the technical writer some hints where to look at and could thus reduce the necessary amount of “reading”.
- *Retrieval of Technical Documentation:* A related, though not primarily interesting scenario is the search for information within a large collection of technical documents. When a searcher is not able to clearly specify his information need or would rather like to get familiar with some of the system’s properties and possibilities, a document map could offer an extension for query interfaces, thus combining browsing and searching.

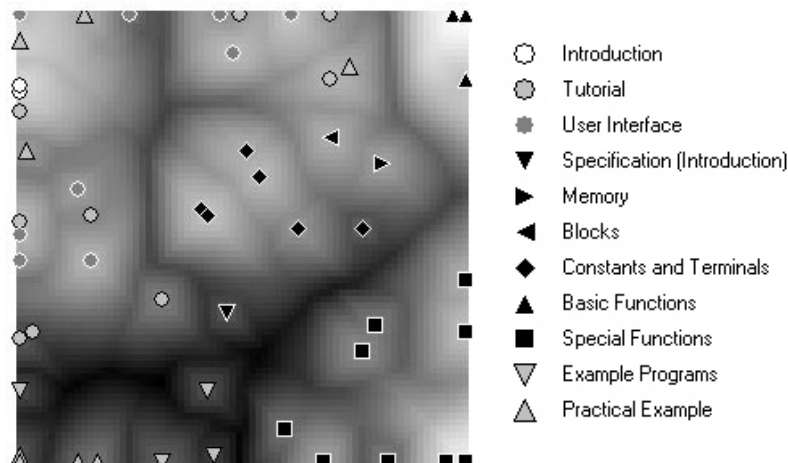


Figure 8-12: Map of user manual sections of a logic module, legend of symbols

8.2.2 Checking the Quality of User Manuals

This section is concerned with analyzing given user manuals regarding the semantic coherence of chapters and sections. The first example tackles the question whether document maps yield a sufficient basis for supporting quality assurance of user manuals. More precisely, the semantic structure as presented by a document map is compared to the intended semantic structure as desired by the authors of the documentation. The second example sketches how document maps can be used to improve user manuals and to define single sources. The reported work (including a preceding evaluation of the maps' quality) has been performed in co-operation with a local technical consulting agency. All user documentation provided for this work is completed and released material. As a consequence, an active improvement of working procedures cannot be expected. Rather, the focus of this section is on the question whether the proposed document map methodology can provide meaningful insights into the user manuals' semantic structure and potentially improve the material with respect to the QA aspects as discussed above.

8.2.2.1 Checking the Semantic Structure of a User Manual

In this small application example the semantic structure of a user manual is analyzed. The following presentation is based on the scenario of checking if the semantic structure of a manual reflects the authors' intentions. The aim of this application is twofold: First, it will be shown that the document map approach yields a suitable basis for the QA task. Second, it will turn out that the map can serve as a means to critically review the structure of a given user documentation.

The user manual used here describes a universal logic module (a programmable control circuit) for small-scale automation tasks which occur in households but also in industry and engineering. Besides the module's specification the manual presents a simulation software which supports the creation of control programs. The manual consists of 5 main chapters: The first chapter briefly introduces the product. Before a stepwise description of the logic module and its functions in hard- and software, a tutorial provides information for getting started quickly. For didactic reasons this presentation is done using concrete examples. After that the user interface of the simulation program is presented. The main chapter gives a detailed specification of the programmable logic module (memory, blocks, constants and terminals, basic and specific functions) before example applications are discussed which are intended to consolidate knowledge about the system's usage.

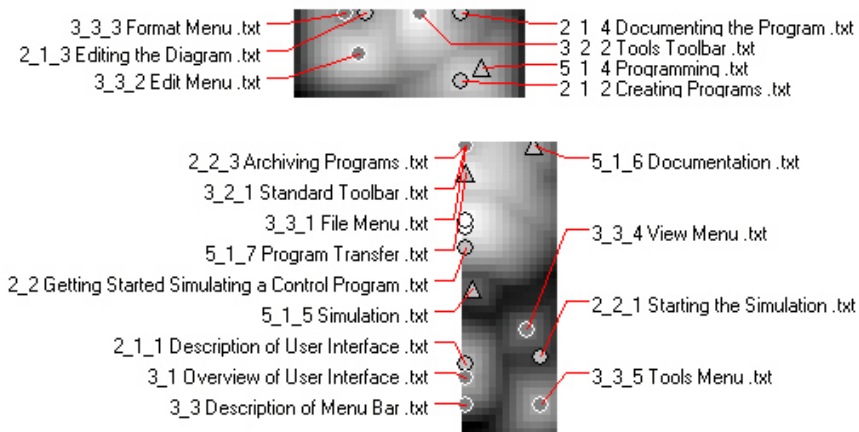


Figure 8-13: Details of the logic module map. Center of northern edge (top) and north-western part (bottom)

Figure 8-12 shows a document map of this user manual. For the map generation the original user documentation was broken down into 63 subdocuments. Each subdocument consists of a heading (up to heading level 3), immediately followed by the corresponding text body. Headings followed directly by subheadings have been removed. Table 8-4 summarizes the main characteristics of the collection and the parameters chosen for map calculation. In figure 8-12 the respective documents are color-coded regarding the original manual chapter they belong to. The quality of the map has been evaluated by the technical writer who was responsible for the documentation project. It turned out that the map faithfully reflects semantic relationships between the single text sections.

Given that it is safe to assume that the quality is sufficient, what insights into the semantic structure of the manual can be gained? First it can be seen that the specification sections of the documentation are coherently distributed among the map. More precisely, there are 3

Table 8-4: Parameters and statistics of collections and map

collection	name	Programmable Logic Module	Steel Casting Simulator
	# documents	63	60
	avg. # words	251	151
indexing	size of vocabulary	847	643
	avg. # indexing terms / document	48	38
	weighting scheme	TF	TF
	similarity measure	cosine	cosine
doc. space	dimension	25	25
	avg. stress	0.089	0.035
SOM	size of SOM	70 × 70	70 × 70
	training steps per document	25	25
	total computation time	39 sec	36 sec

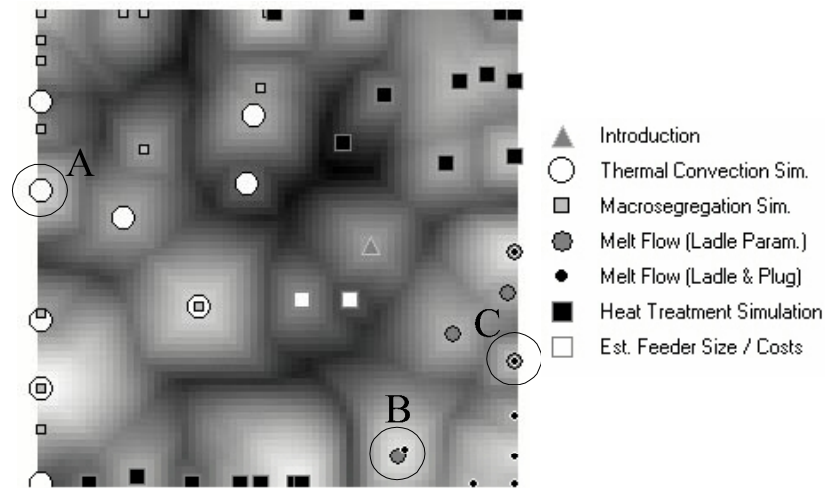


Figure 8-14: Map of user manual sections of a steel casting simulator, legend of symbols.
The document points marked as 'A', 'B' and 'C' represent single source candidates

neighbored areas in which these documents can be found: The south-eastern corner contains documents describing special functions of the logic module, the north-eastern corner comprises all basic function descriptions. Documents about constants and terminals to be used with the model are located in the center of the map. The introductory section of the specification chapter is rather isolated by dark borders. However, it is located near the specification group. The map shows that the specification sections form more or less 'closed' parts of the overall document. Since the specification part serves as a reference manual this isolation is intended. Because the didactic concept of the manual is to allow a quick introduction to the product by presenting a practically oriented tutorial there should be a clear semantic relationship between the description of the user interface, examples and the tutorial itself. Indeed, the map shows that these relationships exist: Taking a closer look to the north-western part and the northern edge (cf. figure 8-13) – besides the two documents of the manual's introduction – the tutorial, the user interface description, and some "practical example" texts can be found. The titles of each respective section indicate that these documents are semantically related. The semantic structure of the manual as presented by the document map can be seen as a confirmation that the user manual has indeed the structure which was intended by the authors. At the same time this is a good validation of the quality of the map approach.

8.2.2.2 Using a User Manual Map for Quality Assurance

The second application example is concerned with a more detailed analysis of a technical user manual's quality. The user manual used in this case describes a steel casting simulator. Such a simulator is used to support product and process planning for steel casting. After an introduction to the product the documentation presents a description of the physical simulation tools in the casting process: estimating the necessary size of the feeder for the casting process, simulating thermal convection during solidification and macrosegregation within the melt, calculating the melt flow rate based on the characteristics of the foundry tools 'ladle' and 'plug', and influencing the structure of the steel by heat treatment. Finally, a tool for assessing the production costs based on geometric characteristics of the casting is presented. The map in figure 8-14 shows the computed structure of the manual. The method used for map generation is identical to that sketched above.

To better understand the relationships between the tools presented in the manual some background knowledge about steel casting is necessary: First, during solidification of the steel a fluid flow can be observed due to high temperature differences in the melt. This flow is called

‘thermal convection’ and causes concentration differences in the casting product, called ‘macrosegregation’. A second important information is the following: In order to simulate the filling of the mold (i.e. the ‘form’ into which the liquid steel is filled in) it is necessary to describe how the melt enters. This can be done by considering the characteristics of the ladle alone or by additionally defining a discharge rate of the ladle based on a plug.

Following these considerations, the respective chapters of the manual are semantically closely connected. By inspecting the map on a general level these main semantic relationships can be redetected: Sections concerned with simulating thermal convection and macrosegregation can be found within the same area (figure 8-14, western part of the map). Some of the respective documents even fall together. This is due to the fact that the contents of these documents is very similar or even identical: The single working steps for these closely related parts are nearly or even exactly the same. Interviews with the technical writer who produced this manual revealed that the respective chapters originally should belong to a single chapter but have been divided for practical reasons: Users who have to perform either a thermal convection or a macrosegregation simulation would like to find all relevant information in one chapter each. However, from the perspective of generating and maintaining the documentation this might impose a source of inconsistency.

Checking document points which are located very close together should help to identify such critical text segments – either to check their consistency or to define a single information source. For example, consider the document point marked with ‘A’ in figure 8-14: This point represents two documents, namely “Interdendritic Flow in the Mushy Zone” and “Permeability in the Mushy Zone”, which both originate from the “thermal convection simulation” chapter. Checking the respective documents shows that their content is nearly identical. There is only a slightly different phrasing in two sentences with exactly the same semantics. The redundant description is, again, intended: The first occurrence of this fragment is meant as an introductory text, describing a process in general. The context of the second occurrence is a description of actions necessary to perform an appropriate simulation of this process. For didactic reasons the general process description is repeated. However, the two text fragments are not derived from a single information source but have been manually created (which explains the slight variation). Given the intention to define single sources, these two text fragments would be an obvious example.

Turning to the group of melt flow documents (south-eastern corner) we find that it is clearly distinguishable from the other groups. Again, this section of the map contains two ‘mixed’ original groups. The semantic justification for this has already been discussed above. Taking a closer look reveals that there are three very close or even overlapping pairs of document points which means that the contents of the corresponding documents seem to be identical or at least highly similar. Consider only the document pairs marked with ‘B’ and ‘C’ in figure 8-14. The ‘B’ points contain documents titled “ladle geometry” from chapter 5.3 and 6.6, the points marked as ‘C’ represent two text segments called “inlet geometry” from sections 5.4 and 6.7 of the manual. Whereas one “ladle geometry” document is an extension of the other, the two “inlet geometry” documents contain the same text with slight changes in phrasing due to the manual re-creation. Like before, this information can be used to define single sources.

Finally, it is interesting to note that the group of “heat treatment” documents is torn apart: Most documents of this chapter can be found in the north-eastern corner, the remaining segments are located at the south-western edge. A closer analysis reveals that most of the south-western documents deal with controlling the heat treatment process by certain parameters. These parameters are all concerned – directly or indirectly – with the temperature of the melt. However, there are no keywords within the relatively short texts which indicate a relation to rest of the heat treatment group. This is a clear problem of keyword-based text analysis: The

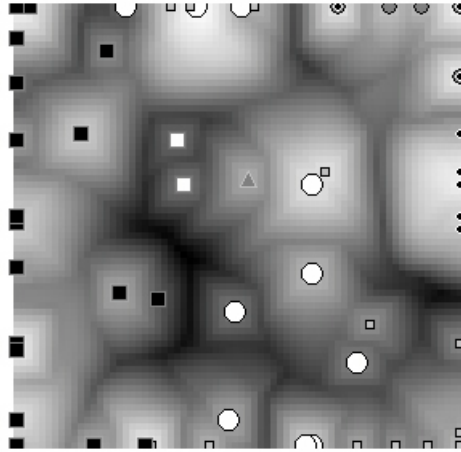


Figure 8-15: Steel casting map after text supplementation

semantic relation between ‘heat treatment’ and ‘temperature control’ has not been detected (in chapter 10 a rule-based semantic refinement component for document maps will be introduced which can help to overcome such problems by incorporating background knowledge).

However, the ‘faulty’ map has resulted in an interesting improvement of the steel casting manual: Analyzing the purely statistical map the author of the manual found that the description of the temperature control segments could indeed benefit from a slight change. The respective text fragments only talk about controlling the process steps but do not explicitly mention the kind of process to be controlled. Though this information becomes quite clear from the context, from a didactic and stylistic point of view it is desirable to have this kind of redundancy. The reason is that a reader who only has a glance over the text should be able to immediately understand the context. Thus, the text fragment “(‘harden’, ‘quench’, or ‘temper’)” – specifying the possible processes to be controlled – has been supplemented behind the term “process step” which occurs in the 4 sections of the manual. The resulting map is shown in figure 8-15. The ‘completed’ documents are located in the north-western corner of the map, now within the remaining text sections, but still slightly separated by a dark border.

8.2.3 Inspecting a Complete Product Documentation Collection

The last section has shown that the document map approach yields a solid basis for checking the semantic coherence of a user documentation’s chapter and section structure. These experiences, however, were only based on relatively small collections derived from single user manuals. In an extension of this case study a complete collection of foundry simulator documentation including the manual of the steel casting simulator already known from above has been examined. This text corpus consists of manuals for all available simulator modules along with the respective module descriptions, online help texts, manuals for options which provide additional functionality, and tutorial documents. Similar to the procedure chosen for generating the user manual maps in the last section, each document has been divided into smaller, semantically coherent sections. In agreement with the responsible technical writers this was done with respect to the section structure and the granularity of each document. Details of the collection and the computed document map can be found in table 8-5.

Figure 8-16 shows the resulting document map. Each document is assigned to one of the groups ‘manuals’, ‘tutorial’, ‘online help’ or ‘online help glossary’ (the meaning of the document symbols is given in the legend below the map). An intensive discussion of the map’s contents would be beyond the scope of this chapter and would reveal only limited additional

insight. Thus, the following considerations are restricted to some exemplary observations which illustrate the value of the document map approach for inspecting a complete collection of product documentation texts.

As for the user manual maps from above the quality of the document map has been evaluated and judged as semantically sound by the responsible technical writers: Documents which are concerned with a particular software module or a certain feature are indeed located in common or neighbored areas (e.g. documents belonging to the steel casting module can be found in the central and south-western part of the map).

What are the insights into the complex collection that can be gained by the document map? The map is particularly interesting when it comes to searching for possible starting points to further improve the documentation material. Consider the document map in figure 8-16: Documents belonging to the different user manuals are scattered among the complete map. This is what should be expected since the manuals cover the complete spectrum of information available for the foundry simulation software. Within a certain surrounding of most manual documents there are also online help articles which describe important system functions. Most online help documents are not mapped to the same point as the corresponding manual document. This is due to the fact that the online help has been written separately; it has not been generated directly on basis of the manual documents. Following the discussion from section 8.2.1 these highly correlated documents could be condensed and defined as single information sources.

Besides serving as a visual interface to search for single-source-candidates, the map can help to find ‘documentation gaps’ which could be filled. The following observations can be derived by figure 8-16:

1. It can be seen that the online help coverage is not complete, i.e. there are possible starting points for providing additional online information.
2. The product’s online help contains also a glossary of important terms. The distribution of corresponding glossary entries in the map reveals for which ‘areas’ glossary entries are defined and helps to find starting points for writing further entries.
3. Consider the distribution of tutorial documents: Tutorial sections are concentrated mainly in the north-eastern part of the map which deals with the general usage of the system. Since the tutorial provides information on getting started with the product this is reasonable, too. If, however, it is desired to extend the manual, the map could be used to identify topics which could be covered by additional tutorial sections.

Using the document map as a visual aid to identify possible ‘information gaps’ as starting points for additional work has been judged as very valuable by technical writers.

8.2.4 Conclusion

This case study has shown that the map contains rich information about the collections’ structure which can directly be used for tasks concerned with condensing documented knowledge and checking the quality of user documentation. The document map approach has been successfully applied to find several starting points for defining single sources. Furthermore, possible ‘information gaps’ to be filled could be easily identified. In section 8.2.2.2 the obvious separation of a semantically related group of documents (“heat treatment”) has lead to a closer analysis of this group. As a consequence it turned out that from a stylistic point of view the coherence of the respective sections should be made clearer in the presentation – a good anecdotal evidence for the value of the approach.

It can be concluded that the proposed document map approach can provide valuable insights into a complex collection of user documentation. According to the technical writers and consultants who participated in this work, the added value of the map approach is mainly the intuitive visual presentation which allows a goal-directed reading of text sections, thus saving time and reducing mental effort. Though it would be interesting to perform a field study in which the document map approach is applied during a complete life cycle of product documentation, the experience reported here should impart a good understanding of the maps' value for visually managing technical product documentation.

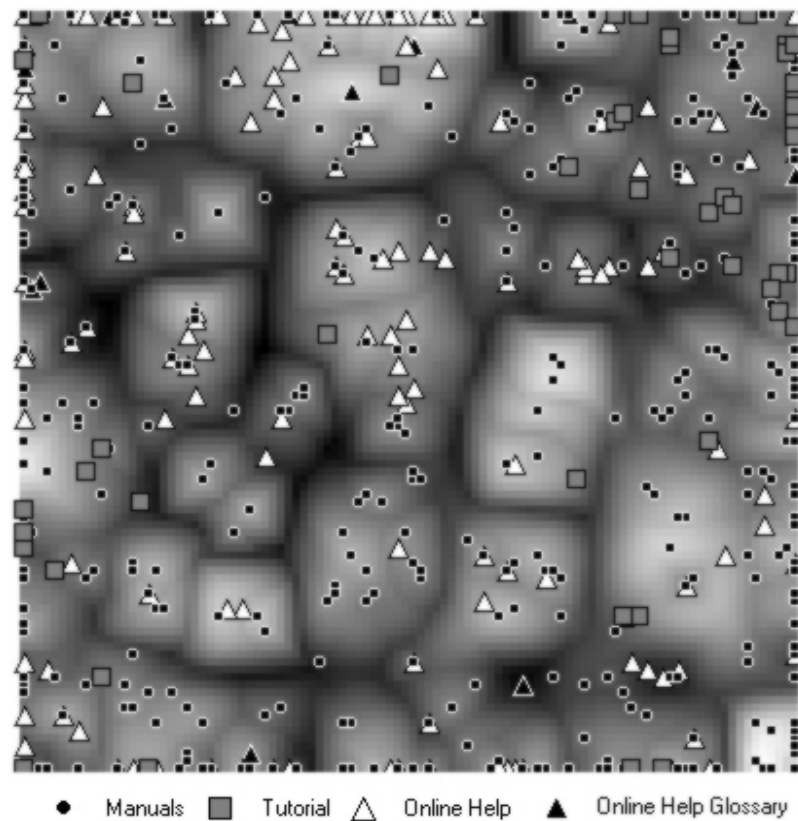


Figure 8-16: Document map of a complete collection of product documentation sections

Table 8-5: Parameters and statistics of collections

collection	name	Complete Foundry Simulator Documentation
	# documents	679
	avg. # words	225
indexing	size of vocabulary	2425
	avg. # indexing terms / document	38
	weighting scheme	TF
	similarity measure	cos
doc. space	dimension	50
	avg. stress	0.35
SOM	size of SOM	100 × 100
	training steps per document	20
	total computation time	22 min 41 sec

8.3 Case Study III: Supporting Terminology Work in a Cultural Science Project

Terminology work is of high importance for the progress and quality of cooperative projects. This is especially true for large-scale interdisciplinary research efforts like the collaborative research center “Media and Cultural Communication” (FK/SFB 427 “Medien und kulturelle Kommunikation”) which studies, among other issues, the impact of multimedia information systems on cooperation and organization of knowledge in cultural science projects. In this highly interdisciplinary context it is crucial to understand the ‘language’ used by different partners. For this reason it is an important goal to set up a ‘research center dictionary’ that defines the ‘terminological culture’ of the project.

Cooperative terminology work is a process that consists of several phases. In early phases it is important to identify relevant terms and concepts which can be discussed and examined in the further work – a step that can be supported by collaboratively analyzing existing documents. This case study works out and evaluates a method for supporting such an analysis using the proposed document map approach. The method – cooperative document analysis with document maps – is integrated into the methodology of iterative and cooperative terminology work in a natural way. By means of visualizing collections of research center documents and interacting with the resulting document maps, all participants are included in the cooperative analysis – despite different professional backgrounds. The proposed method could clearly press ahead terminology work in the cultural science project “Media and Cultural Communication”.

This case study differs in some significant ways from the others: First, whereas the preceding case studies rather had lab-character (i.e. they deliberately have not affected ongoing work, but nevertheless have been performed within a real application context) this section describes a ‘field experiment’: The results of working with the map are used directly in a large project context. Second, unlike in the other cases, not only one or two analysts work with the document map. Rather, the analysis work is performed in a highly cooperative manner where the document map serves as a means for supporting communication. Third, the document collection analyzed here does not contain technical documentation but texts with a completely different character, namely essays and papers from an interdisciplinary cultural science project. This work has been published in [BK01].

The remainder of this chapter is organized as follows: The next section provides some background on the cultural science project and the context of this case study. Section 8.3.2 introduces a method for cooperative document analysis with document maps and embeds it into its research context. After that, some results achieved by applying the method will be sketched and the approach will be evaluated.

8.3.1 Knowledge Organization in a Cultural Science Project

Creative research and development processes in cultural sciences are increasingly characterized by cooperation and interdisciplinary work. In this context an important task of computer science research is to provide adequate information systems for effectively supporting these processes. Hence, an essential element for defining requirements and finding suitable solutions is to analyze courses of communication, coordination, and cooperation as well as organization of knowledge in those creative processes in an information-system-oriented way.

The interdisciplinary research center “Media and Cultural Communication” (SFB/FK 427 „Medien und kulturelle Kommunikation“) partly devotes itself to this task. Furthermore, the project itself is subject to an intensive analysis of information flows since its highly interdisciplinary and collaborative structure provides valuable insights into creative research and development processes. Within the research center such processes are intensively examined and – based on the analysis results – cooperative information systems are introduced.

This chapter focuses on cooperative document analysis as part of a general methodical approach for supporting cultural science work. The document analysis is based on texts which have been written by the participating scientists and which are specific for this cultural science project. The goal of cooperative document analysis is to identify ‘terminological cultures’ by extracting a common and basic terminology from the produced documents, thus creating a basis for a systematic terminology work within the research center. Before presenting the method for cooperative document analysis with document maps it is helpful to briefly sketch the central research questions of the collaborative research center “Media and Cultural Communication”.

8.3.1.1 Collaborative Research Center “Media and Cultural Communication”

The changing book-culture in the context of a general change of media is the central starting point of the collaborative research center “Media and Cultural Communication”. This research network is a cooperation between the universities of Cologne, Aachen and Bonn and employs about 30 scientists from fields like linguistics, literature, art history, psychology, neuro-linguistics, ethnology and computer science. All those experts work closely together in 13 subprojects which are all concerned in the following general questions:

- How do differences of media affect the world-wide culture in past, present and future?
- What are the effects of the change of media on cultural communication in society and cultural activities?
- What have been the methods of media research in past, and how can they be improved?

In particular, the development and usage of cooperative information systems (like the BSCW system [BAB+97], cf. section 8.1) are examined as a new way of communication among scientists. This research center offers the first opportunity to intensively analyze effects of media in a large scale and in an interdisciplinary environment.

8.3.1.2 Cooperative Information Systems in Cultural Science

In the research center a central task for computer scientists is to analyze existing information flows within cultural science projects in a media-specific way. Moreover, information systems have to be provided which promise to improve the support for cooperative work in the future. For the task of analyzing information flows the meta-information system MAVIS [FKM00] has been introduced. This tool offers a means for self-description by enabling participants of cultural science projects to describe patterns of cooperation and knowledge organization and to compare them with other existing patterns [KlJa99]. Besides, methodical experiences with cooperative information systems which have been successfully applied in engineering projects [NJJ+96, JaKe99, KPJ00] can be applied for cultural sciences (see also sections 8.1 and 8.2).

8.3.1.3 Media Analysis in the Research Center

Studying information flows in a media-specific way requires an extensive understanding of the application of media. Therefore, a complex media analysis has been performed in several steps. In a first step, cooperation processes and knowledge organization in cultural science have been roughly analyzed based on literature reviews. The analysis revealed a multitude of cooperation processes, for example cooperative writing, reading and edition, design of exhibitions, and others. Consequently, for a more detailed investigation the focus has been set to a specific process, namely cooperative development of terminology. Cooperative terminology work is a complex process which can be studied regarding the traditional application of media as well as concerning the effects of media changes. Furthermore, terminology work is important for developing and stabilizing interdisciplinary and cooperative methods in the collaborative research center.

For this particular process a detailed conceptual analysis of information flows and application of media has been performed. In group workshops the information flows of large subprojects have been analyzed in a working place oriented way (cf. [NJJ+96, JaKe99]). Each subproject has produced an interior and exterior view of its information flows. By doing this, information channels, applied media and quality of information flows could be determined. The major finding of this step is that many information flows take place on an informal level and depend on the presence of the respective communication partners. In order to support information flows in the context of terminology work it is therefore necessary to introduce information systems which guarantee the maintenance and persistence of communication.

In a third step different systems for supporting cooperative terminology work have been presented and discussed by the researchers, the goal being an iterative development, test and evaluation of an adapted ‘work bench’. It turned out quickly that dedicated systems for terminology work can be applied only when terminological cultures within the project can be identified. This can be done by extracting a set of relevant terms from existing electronic documents of the research center (e.g. essays, conference announcements, protocols of colloquiums and seminars). These terms would then introduce the more formal and structured part of cooperative terminology work.

This is where the document map approach comes in: Experiences from technical disciplines show that document maps promise a valuable support for condensing knowledge which has been elicited by distributed and heterogeneous working groups (cf. sections 8.1 and 8.2). By working out the current state of the ‘terminological structure’ of a collection of relevant publications the initial step of terminology work could be supported. In the following, a method for supporting the extraction of relevant terms based on document maps is presented. Some results of the method’s application are discussed in section 8.3.3.

8.3.2 Cooperative Document Analysis with Document Maps

8.3.2.1 Objectives of Cooperative Document Analysis

As indicated earlier, cooperative document analysis is an important part of terminology work. The probability that two persons use the same term when they mean the same thing is less than 20% [FLGD87]. This phenomenon – known as the vocabulary problem – is a significant obstacle in project-oriented and interdisciplinary work. Thus, a crucial factor for defining a useful terminology is that all participants develop a common understanding of concepts and arrive at a collective agreement about the meaning of terms to be defined. Consequently, a discussion about the usage of terms within the different disciplines and subprojects is manda-

tory in the context of the cultural science research center. A lack of terminological culture would result in serious difficulties in reading or discussions, since terms which are familiar in a certain discipline may be misunderstood in the context of another discipline. Furthermore, the transfer of a term's meaning would be complicated.

Thus, a cooperative procedure in terminology work is a decisive factor of success, especially in standardizing terminology work (cf. DIN 820 [DIN77] and DIN 2339/1 [DIN86]). Approaches from cognitive psychology [Quil68, Mins75], philosophy and logic [BrSc85], and linguistics [Fill68, Sowa84] concentrate on problems of representing a terminology. Though information system approaches have evolved significantly, e.g. WordNet [Mill95], Ontolingua [FFR96], or ConcepTerm [BoFa94], cooperative aspects are widely neglected. Furthermore, existing approaches mainly support later phases of terminology work [Buol99] or provide an adequate infrastructure for information and document exchange and virtual group meetings (cf. BSCW system [BAB+97]). Such systems contribute to group awareness by providing a cooperative, Internet-based 'writing desk' which increases the transparency of group events. The terminology server 'concept' [BKJJ97, Buol99] supports three aspects of cooperative terminology work: For one, the process itself (as proposed by DIN 2339/1 [DIN86] and ISO 10241 [ISO92]) is guided. Second, the system allows to define and automatically check project-driven or requirement-driven quality aspects for a terminology. Finally, 'concept' contains a component for supporting cooperation which reports terminological developments (e.g. the addition of a synonym to the terminology) to the project participants according to adjustable interest profiles.

However, the spectrum defined by these tools leaves some space for further support. In early phases of terminology work neither a suitable structure of the taxonomy to be defined nor a sufficiently extensive set of coordinated concept descriptions is available. Moreover, in the context of the interdisciplinary cultural science project described here, the problem arises that many scientists are not used to formal computer science or mathematical methods. Therefore, an explorative and iterative approach to the formal definition of a terminology in a later phase seems appropriate in order to lower the threshold for the participants. From a methodical point of view the current usage of terms (which possibly leads to difficulties of understanding) has to be examined and discussed. Cooperatively analyzing persistent documents is a good starting point since the texts record the current term usage and are suitable for a statistical analysis.

8.3.2.2 The Role of Document Maps in Cooperative Document Analysis

A graphical presentation of a document collection's structure – as realized by document maps – allows an explorative and iterative access to the material and promises to promote the discussion between the different disciplines. In order to effectively support cooperative document analysis it is necessary to develop a method for applying document maps in the context of interdisciplinary terminology work. Here, special emphasis is given to the need for providing insight for the researchers into the usage of terms in different scientific fields. More specifically, it is necessary that scientists understand the contexts in which certain terms are used and that they recognize the differences in the meaning of these terms. Thus we need an explorative and cooperative way to understand the structure of term usage.

The document map system DocMINER (cf. chapter 7) promises to be a suitable technological basis for extracting term usage from text collections as it allows to analyze document relationships on a fine-granular level. Moreover, its interactive features allow to explore textual relationships step-by-step. A document map of the collection under consideration potentially offers a common basis for exploring the current terminology and promises to act as a promoter for discussions. However, in order to effectively apply such a system in the highly coopera-

tive context of terminology work, it is necessary to define a suitable method of operation. Based on the sketched profile of requirements and experiences from the technical case studies (sections 8.1 and 8.2) a method of cooperative document analysis with document maps has been developed. In the context of the cultural research center the method – which is sketched in the following – has been presented and refined in a preparatory group workshop.

8.3.2.3 Document Map Application in Terminology Workshops

Starting point for applying DocMINER in terminology work is a document map which presents the similarity structure of a corpus of texts based on term usage (more details on that are given in section 8.3.2.4). Given such a document map, it is important to find an adequate procedure for document analysis. Due to the cooperative nature of terminology work, moderated group sessions are proposed in which participants of a working group collectively analyze the map and discuss the findings. The document map system is operated by the session leader. The system's interface is visible to the group by a beamer projection. The analysis itself is organized mainly in 3 phases:

- *Analysis of the visual map presentation.* A first step is to examine which subgroups (e.g. subprojects) are terminologically coherent or which 'groups of term usage' do exist. Therefore, the visual grouping of documents in the map is analyzed: Groups are determined and their semantic relationships are worked out intellectually. On this level it is possible to look for 'bridging documents' which associate different groups with each other by their term usage. In the map, these documents can be found between neighbored groups.
- *Determining relevant terms for groups and documents.* Term profiles of groups are a starting point for discussing concepts and term usage in the context of cooperative document analysis. For the groups identified in the first step statistically significant terms are computed (cf. chapter 7.4.2). From the resulting lists of terms those items are selected which are scientifically relevant for the group under consideration. Such terms are possible elements for setting up a formal terminology. The meaning of these terms and their usage in the given context are explained by the respective authors and can be discussed by the group.
- *Cross-check.* Since term profiles do not strictly differentiate groups, some terms may play an important (and possibly different) role in other groups, too. Finding documents which use a certain term more or less frequently can be supported by the query interface of DocMINER (cf. chapter 7.4.3). The visual display of the query result in the document map can help the group to quickly grasp term frequency and the context of single terms. Highlighting documents in the map display which also use certain terms is an essential support for a goal-oriented discussion.

Even though these steps should not be seen as strictly consecutive, they do support a structured use of the document map tool. This is of particular importance since such workshops are characterized by vivid discussions and experimentation with the map, as experience shows.

8.3.2.4 Acquisition and Preparation of Documents and Document Map

The documents which have been produced in the cultural research center (publications, applications, conference announcements, protocols) determine the usage of terms and are thus the basis for cooperative document analysis and terminology work. Due to the heterogeneity of subprojects and the interdisciplinary nature of the project as a whole, a first and basic task is to identify subprojects which share terms or are rather 'isolated' regarding term usage. Fur-

thermore, the usage of relevant terms within different disciplines and subprojects has to be discussed and understood.

Regarding the generation of a suitable document map, a first question that arises concerns the level of document granularity: Which documents or parts of documents form a relevant contribution that should appear as an independent text on the map according to the given analysis task? Since most documents (in particular essays and applications) do not form monolithic papers but rather consist of topically coherent parts the texts can be subdivided into sections. Having discussed this question in the group, each paper has been subdivided according to its topical parts based on the respective author's judgment. Footnotes – which mainly contain bibliographic notes – have been deleted completely since they do not reflect the current usage of terms. The same holds for lists of references. Citations – especially those in foreign languages – have also been removed. The single documents have been named in a canonical way so that they can be easily assigned to their respective context. Besides a descriptive title, each document name contains a code for its subproject and for the main document it has been derived from. The authors used the BSCW system to file the prepared documents in a special public work place so that the resulting document collection could be inspected by all project participants. This procedure offered a transparent document exchange within the terminology working group and was particularly helpful for getting a consistently prepared collection of texts for document analysis.

Since the document map was intended to present similarities based on used terms, the keyword-oriented vector space method of document representation and comparison was applied (cf. section 6.2.1.3). A thesaurus was deliberately not used since it would have anticipated results of terminology work and would possibly have formed an undesired bias. Up to the completion of this thesis, two workshops have been performed. A third one is scheduled which will use an extended collection of text material. Details of the collections and map parameters can be found in table 8-6.

Table 8-6: Parameters and statistics of collections

collection	name	workshops 1 & 2	workshop 3
	# documents	128	280
	avg. # words	764	812
indexing	size of vocabulary	11,859	20,083
	avg. # indexing terms / document	279	291
	weighting scheme	TF × IDF	TF × IDF
	similarity measure	cosine	cosine
doc. space	dimension	100	180
	avg. stress	7 %	10.6 %
SOM	size of SOM	100 × 100	120 × 90
	training steps per document	20	25
	total computation time	7 min 14 sec	24 min 16 sec

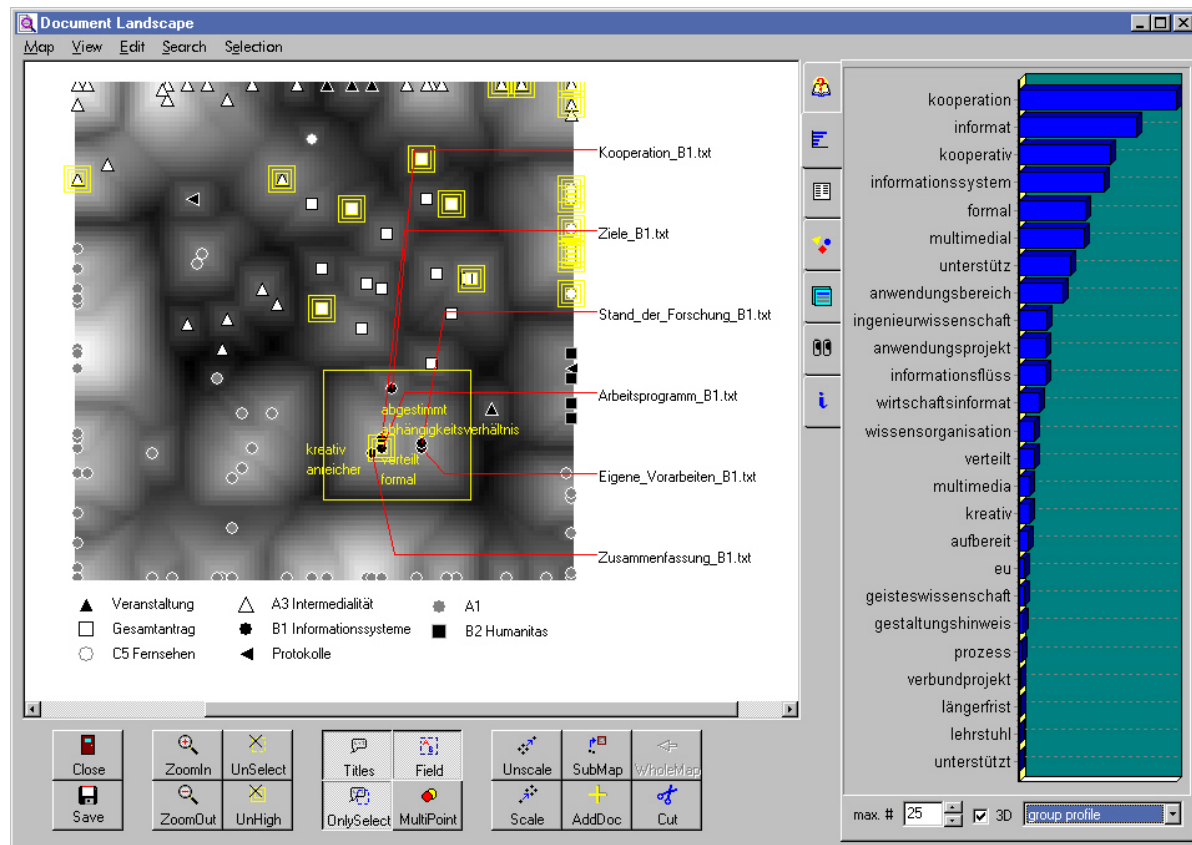


Figure 8-17: Document map of cultural science publications within the DocMINER interface. The legend of document symbols is given below the map. The right hand side shows the term profile of the marked group. Highlighted documents in the map match the query term “Fernsehen” (“television”).

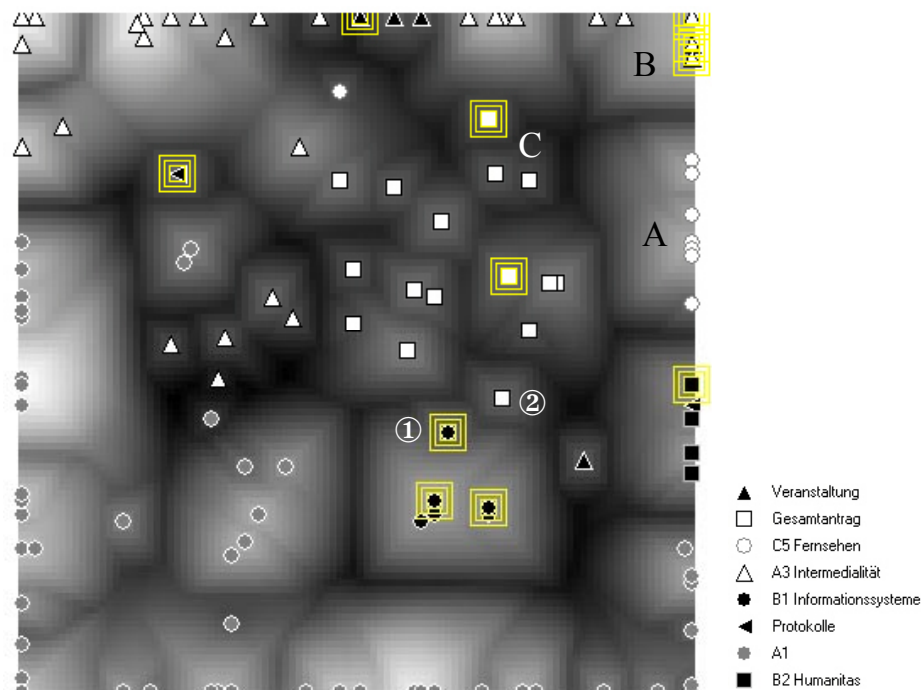


Figure 8-18: Magnification of the cultural science map. Documents and areas which are discussed in the text are marked. Highlighted documents match the query term “hypertext”.

8.3.3 Some Results of the Cooperative Document Analysis

The cooperative document analysis within the cultural research center was performed in two workshops up to now (August and October 2000, 2 hours duration each). In each workshop about 15 scientists participated. At the beginning of each session the method for working with the map was repeated briefly. The central part of each workshop was the cooperative discussion of the collection's structure as conveyed by the document maps. To initialize the discussion, term profiles of selected groups were presented and explained. Figure 8-17 shows the document map of a collection of 128 (sub-)documents (mainly application documents) and the term profile of the marked group. Figure 8-18 presents a magnified version of the same map with additional markings.

As a first result it can be stated that documents that were divided into small collections of subdocuments can be found in coherent or neighbored regions of the map. The general application for the collaborative research center (in German: "Gesamtantrag", cf. legend of symbols in figure 8-17 or figure 8-18, respectively) shows a rather heterogeneous structure: The jagged area in the map indicates a variety of topics. This complex document was written by representatives of all subprojects and represents all aspects of the project as a whole more or less intensively. The group's term profile contains generic terms like "communication", "interface", "medium", "media theory" or "technology" (in German: "Kommunikation", "Schnittstelle", "Medium", "Medientheorie", "Technik").

A second interesting observation is the following: The application for a subproject which joined the research center only in a later phase uses a terminology similar to that used in the general application (cf. figure 8-18). Document '①' from subproject B1 and text '②' from the introduction to the general application are closely neighbored (center of the map). Important common terms are "networking", "informatics" and "communication" (in German: „Vernetzung“, „Informatik“, „Kommunikation“), the reason being that subproject B1 (which deals with cultural effects of information systems) used an early version of the general application and is represented in the application with own contributions. Moreover, both documents contain text sections which sketch a possible cooperation between subproject B1 and other projects. For terminology work this means that computer science terminology is used in the general application despite a late integration of this discipline.

Subproject C5 is concerned with the medium "television" (in German: "Fernsehen", area 'A' in the map, figure 8-18). The area is rather isolated and shares only few relevant terms with other areas. Characteristic terms of the document group are "media discourse", "broadcasting", "television" or "consumer" (in German: "Mediendiskurs", "Rundfunk", "Fernsehen", "Konsument"). From the viewpoint of terminology work this is an example for a rather independent usage of terms. For setting up a consistent terminology it is important to find the rare documents which use the very special terms in other contexts and to discuss and understand their divergent usage. For example, the central term "television" is used essentially in C5 but also occurs in subproject A3 which deals with building interfaces by technical devices (cf. query result in figure 8-17). However, in this context the meaning of "television" is quite different: The discussion of term usage revealed that C5 uses the term in the sense of a media discourse whereas A3 understands "television" as a medium. This has to be taken into account during the later definition of a more formal terminology.

Another relevant observation is that texts which were written by the same author may use a different terminology. A striking example is given in figure 8-18: Consider area 'B' in the map, which contain a scientific essay, and the three documents of the general application ("Gesamtantrag") in area 'C'. Both groups of documents are characterized by different term profiles. More precisely, the application uses more general terms like "interface" and "me-

dium” (“Schnittstelle”, “Medium”) whereas the scientific paper in area ‘B’ makes use of concrete instances of the general terms, e.g. “writing desk” (“Schreibtisch”) and “Internet”. The discussion revealed the decisions behind the divergent usage of terms.

As a final example consider the query result marked in figure 8-18: An important concept in the cultural research center in the term “hypertext”. This term is prominent in several documents written by different authors as the figure shows. There was even a conference dedicated to this topic. Despite that, the general application only contains the term sporadically. Again, the discussion revealed background information which is important for terminology work: During the final revision of the general application the term “hypertext” was replaced by the more comprehensive term “hypermedia” in order to stress the ‘triad’ of the German terms “Medialität – Intermedialität – Hypermedialität“ (“Medialität” can roughly be translated as ‘connection of media to communication’) which better reflects the topic of project A3.

8.3.4 Evaluating the Support Offered by Document Maps

Besides performing a cooperative document analysis the workshops aimed at evaluating the support offered by document maps. To do this the use of document maps and the tool DocMINER was discussed within the group. More specifically, the discussion was lead by three questions: Are document maps a suitable means to cope with document collections? Is DocMINER a suitable tool for supporting terminology work in cultural science? Is the cooperative analysis of term profiles a suitable means to understand text grouping, text relationships and ‘bridging terms’ that connect groups of documents?

To sum up the results of the discussion, it turned out that the visual presentation of text material could clearly support the process of understanding text collection and term usage. The statistical analysis of documents becomes transparent by the intuitive visual metaphor of the map and the interaction offered by the tool. Linguists and German philologists did not have problems to accept the proposed method of cooperative document analysis with document maps and to take it up as an aid for cultural science work. In particular the visual metaphor of the document map was understood quickly. Relating to the technical concept of ‘document similarity’, literature scientists observed correctly that texts are not only characterized by term usage. However, with regard to the analysis goal the idea of comparing texts by term statistics (as realized by the vector space method) was accepted. Methodically, the iterative and cooperative analysis aided by the document map was acknowledged as a suitable approach in an early phase of terminology work. It was commonly agreed that an explorative access to a collection of research center publications has obvious advantages over purely checking and querying the usage of terms. Especially the interactive and visually appealing way of working with DocMINER produced a remarkable interest in exploring the current term usage.

The main hypothesis of applying document maps for terminology work was that the cooperative analysis of term profiles opens the participants to discussions. The group workshops performed so far could prove this: Terms from the statistically derived term profiles of visualized groups were taken up and examined without problems. The major result of the workshops is a multitude of discussed and commonly understood terms. The goal to set up a terminological culture was reflected in the process of terminology work aided by document maps: Visualizing coherent and isolated text groups turned out to be helpful for realizing the effect that ‘isolated’ term usage has on interdisciplinary and cooperative work. As a consequence the need to find some ‘bridging terms’ which connect terminologically separated groups arose.

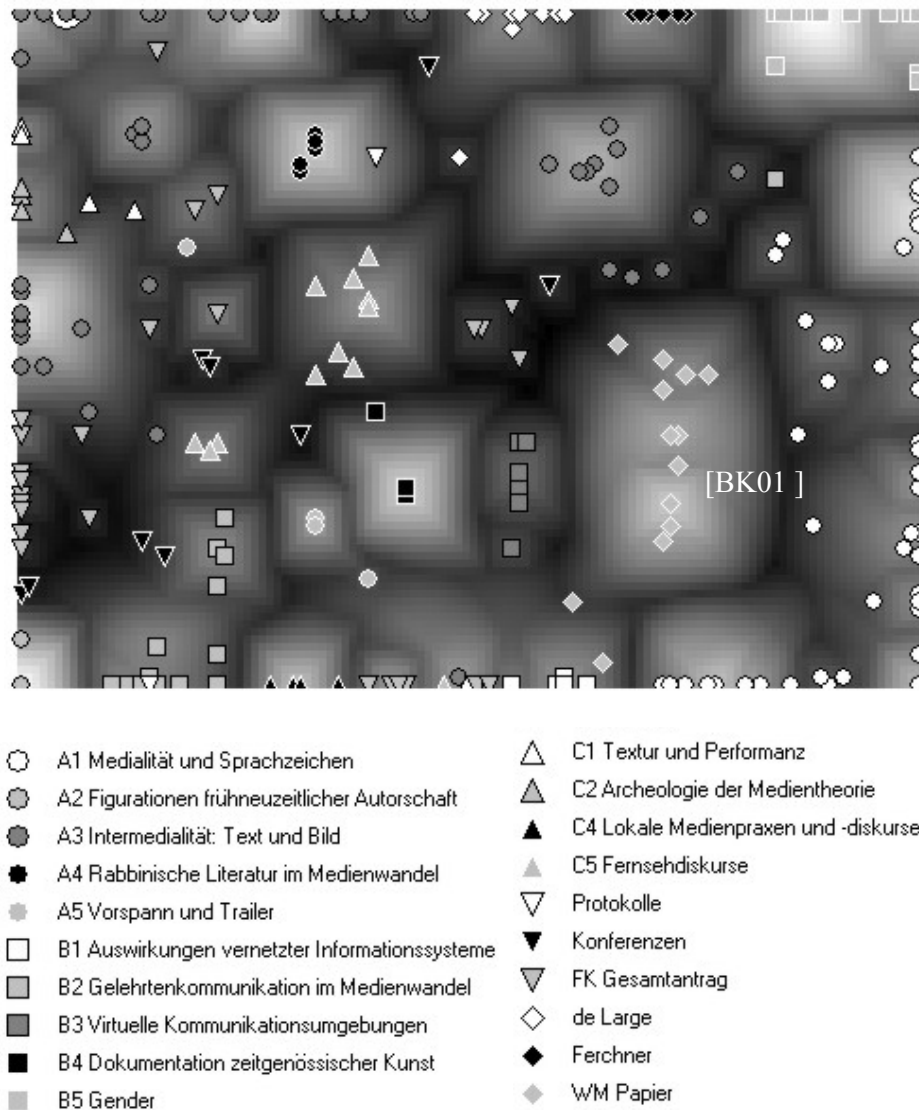


Figure 8-19: Map and legend of extended cultural science collection.
The field labeled with [BK01] contains a German report on this case study.

8.3.5 Conclusion and Outlook

Due to the vocabulary problem cooperative terminology work is of high importance, in particular for large-scale interdisciplinary projects like the cultural research center “Media and Cultural Communication”. Early phases of terminology work are concerned with cooperative analysis of existing documents in order to work out the current usage of terms. This case study has presented and evaluated a method for supporting document analysis by document maps. It turned out that the approach naturally fits into the framework of iterative and cooperative terminology work, the major advantage being that all participants are involved in the process of discussing their understanding of terms and concepts. Visualization and interaction makes transparent the terminological structure of a corpus of research center documents and is appealing to scientists with different backgrounds. Besides evaluating the method as such, the cultural research center’s terminology work could be pushed ahead significantly.

As a result of the intensive and fruitful terminological discussions during the workshops the participants were motivated to extend the collection of documents. As of December 2000 the number of text fragments to be analyzed has more than doubled. The corpus for the third workshop in the series of cooperative document analysis sessions now contains the complete application for financing and several scientific publications. Figure 8-19 shows the map of this extended collection. In particular, a publication on cooperative document analysis ([BK01], a German report of this case study) can be found in the map (named “WM Papier” in the legend).

Decisions regarding the application of the proposed document map approach for the future of the research center stress the positive and encouraging results: The document maps computed so far will flow into the application for the next term of the cultural research center since they demonstrate how novel information systems affect the way cultural science work changes. Results of the map-aided terminology work will be used to bridge the terminological gaps between the general application and the application for subprojects. The aim is to compensate the terminological isolation of subprojects. The development of term usage will be documented in a ‘historic atlas of terminology’: A series of document maps will be used as an additional means of demonstrating the terminological progress of the cultural research center. Furthermore, document maps shall act as a medium for self-observation.

8.4 Summary of Results

Table 8-7 summarizes the characteristics of the main analysis tasks examined in the case studies. For the technical and scientific disciplines sketched in the last sections, document maps turned out to be a valuable support for complex analysis tasks of document collections. In particular, the added value for the analysts is that document maps ease the exploration of complex domain specific text collections: Users can quickly obtain an intuitive overview of the corpus’ structure and gain insight into topical relationships. The results of each individual case study have already been discussed. To state the general findings in a nutshell: For one, the case studies have confirmed that the document map technology proposed in this work produces high quality visualizations of the structure of specialized document collections. Moreover, it has proven true that the approach is highly useful for real-world document collections and analysis tasks: We found anecdotal evidence that document maps improve the way the respective work is done traditionally. Users and project partners say that they can do tasks better than they could before. It also turned out that the approach is transferable to non-technical application contexts: Document maps are not only successful for purely technical and highly specialized text collections (as considered in sections 8.1 and 8.2) but can also be used for rather narrative and stylistically more complex documents (cf. section 8.3).

Beyond considering pure technology, this chapter could contribute to the understanding of how to successfully *use* the document map approach for analysis tasks which are relevant in knowledge management. It has been shown how to effectively exploit the visual information about relationships of individual documents and document groups for tasks like condensing, structuring and synchronizing documented knowledge or supporting cooperative document analysis. Still, there are some important questions open: The case studies yield *qualitative* evidence about the maps’ usefulness. But are document maps significantly and measurably superior to query-driven information retrieval tools for solving the tasks sketched above? Can the experiences from the case studies be generalized? The next chapter is dedicated to a closer examination of these questions.

Table 8-7: Characterization of main analysis tasks from case studies using the task model (cf. chapter 2)

CS III:		supporting terminology work by exploring the term usage in a collection of research papers			
CS II:		check the single source criterion in product documentation			
		check structural quality of user manuals			
CS I:		understand relationships between use cases and refine given a priori structure			
goal of interaction	making use of documents	learn	✓		✓
		condense documents			
		select documents			
	maintaining documents	control quality			
		assure quality		✓	✓
dynamics of focus of interest	fixed			✓	✓
	adaptive		✓		✓
resource considered	information from documents		✓		✓
	meta-information	document attributes	✓	✓	
		structural information	✓	✓	✓
method of interaction	explorative		✓	✓	✓
	goal-directed				✓
granularity	overview		✓	✓	
	details		✓		✓
categories	not relevant				✓
	relevant	use existing		✓	
		use classes	✓		✓
	categorize				
focused relationship	external: document – specification			✓	
	inherent	document – document	✓		✓
		document – topic	✓		
		topic – topic	✓		✓
mode of communication	recognize (reflect, physically passive)		✓	✓	✓
	specify (physically active)				✓

9 Task-Adequacy of Document Maps: A Comparative Laboratory Study

The case studies presented in chapter 8 have qualitatively shown how document maps can be applied effectively for corpus analysis tasks in sample contexts. They yielded anecdotal evidence about the success of the system's usage and indicated that visualizing a corpus' structure adds value to the way analysts work with a collection of valuable corporate documents. The question tackled in this chapter is concerned with a task-oriented quantitative evaluation of the performance of the proposed document map approach, allowing to generalize results on an empirical basis. Therefore, a comparative laboratory study was performed. More precisely, the task model worked out in chapter 2 and results of the industry survey conducted for developing it were used to construct more representative and generalized tasks (in contrast to the specialized tasks examined in the case studies). Using these tasks, the basic concept of the document map approach – the visualization of the semantic structure of a text corpus – was evaluated in a task-oriented way. This was done in a controlled experiment in which two groups of test subjects performed analysis tasks with two systems which differ only in their way to display a collection of documents. The following section specifies the research question and the objectives pursued in the experiment before section 9.2 presents an appropriate experimental design. Section 9.3 sketches the realization of the experiment, chapter 9.4 presents the observations made in the study. Finally, the results are interpreted and discussed. The study was supported by a master thesis [Seel01].

9.1 Research Question and Objectives

In this study the question is tackled whether the proposed document map approach yields a measurable added value for solving document analysis tasks in knowledge management. The core information provided by the document map, that is the visualization of the overall similarity structure of the text corpus, is considered in isolation. More precisely, the main objectives are

- to evaluate whether a computation of the overall similarity structure of the text corpus and its visualization has a significant influence on the effectiveness with which tasks that are relevant in document analysis in knowledge management can be solved,
- to investigate the user acceptance of document maps, and besides
- to collect some data about the way people use a document map system to complete a set of assigned tasks.

In other words, the driving motive is to assess the task-adequacy of document maps in knowledge management on a quantitative level. The system quality is assessed based on criteria that are relevant in the application domain under consideration. The following hypotheses are examined:

- (H₁) A document map is more adequate for the task of *getting an overview of a collection's subject structure* than a comparable system which does not compute and visualize the overall similarity structure of the text corpus: Users of the map will get an idea of the collection's topical structure more effectively than users of the alternative system.
- (H₂) A document map is more adequate for the task of *checking the clearness of the semantic structure of a text collection* than a comparable system which does not compute and visualize the overall similarity structure of the text corpus: Users of the map will identify unusual documents more effectively than users of the alternative system.
- (H₃) A document map is more adequate for the task of *checking the correctness of a classification of text documents* than a comparable system which does not compute and visualize the overall similarity structure of the text corpus: Users of the map will identify classification mistakes more effectively than users of the alternative system.
- (H₄) A document map is more adequate for the task of *understanding the semantic context of documents and associating related texts* than a comparable system which does not compute and visualize the overall similarity structure of the text corpus: Users of the map will associate related documents more effectively and will better catch the contents of each related group in the time available than users of the alternative system.
- (H₅) Users judge document maps to be more adequate for solving the assigned tasks than a comparable system which does not compute and visualize the overall similarity structure of the text corpus.

9.2 Experimental Design

In this section a precise formulation of the hypotheses sketched above and an experimental setting to test them is worked out. In particular, the system support for the experiment is presented, measures for assessing the task-adequacy are defined, practicable tasks are set up which can be performed by test persons and which are open to an objective evaluation. Then, corresponding formal hypotheses are formulated which can be tested in a comparative empirical study.

9.2.1 System Support: Document List and Document Map

The leading objective of this study is to evaluate the value added by computing and visualizing the overall similarity structure of a text corpus for document analysis tasks. A comparative system must thus differ from the document map system exactly in this point so that observed results can be reduced to this aspect. Figure 9-1 depicts an abstract view on the interaction with the map-centered text-access system proposed in this work (cf. chapter 7): An analyst – driven by the assigned task – uses the text-access system to work with the collection of documents. The visual workspace (i.e. the document map) is the central part of the system. It provides an overview of the stored texts. In particular, it presents document symbols and meta-information (e.g. a color-coding regarding certain criteria). To access the collection the user selects documents in the workspace or poses requests (e.g. keyword queries) to the base interface of the system. The results of these requests are displayed in separate windows, but

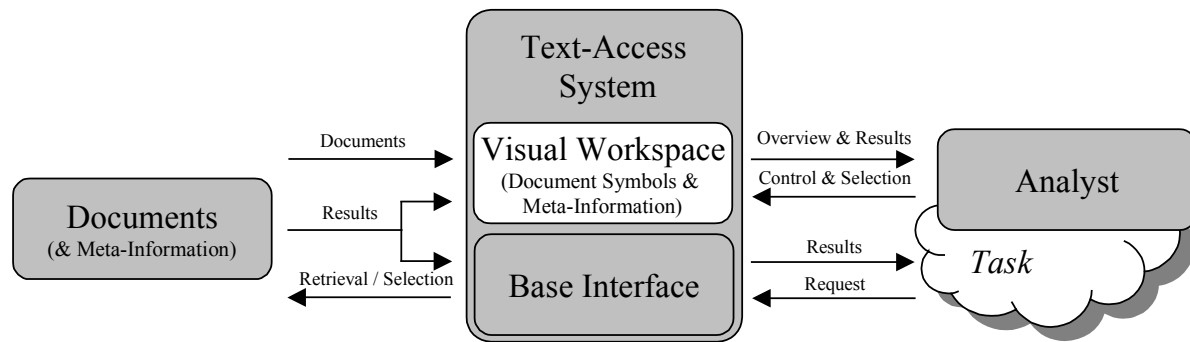


Figure 9-1: Interacting with the text-access system: information and control flow

can also be displayed in the visual workspace (e.g. highlighting of documents that match a query). Consequently, a comparative system has to differ from the document map system only by the *kind* of overview presentation, not by its existence as such. Results of requests must also be displayed in a visual workspace which provides a certain overview and – even more important – allows permanent and direct access to all documents. The visual workspace of the alternative system used in this study is a document *list* which presents the titles of all documents available, along with document symbols which represent meta-information (such as color-coding). Text lists are the most natural kind of alternative representation. The comparative workspace can be operated just like the Windows file explorer.

The systems used in the experiment differ from the basic document map system (cf. chapter 7) in several aspects: First, some advanced system features are removed to simplify the system usage. Both experimental systems comprise the following functions: basic workspace interaction (opening documents on the workspace, selecting arbitrary subsets of documents, viewing highlighted query results and meta-information), query interface with Boolean and full-text queries, group profile of terms, displaying full-texts, interactive legend of symbols (with highlighting of selected classes), calculation and presentation of important sentences of a selection, and displaying a compact list of selected documents. The graphical system interfaces cover the entire screen and cannot be minimized.

The interfaces are embedded in an experiment control system which supervises the course of actions: It loads necessary data, displays or hides meta-information for each assigned task (e.g. a given classification in the legend window is only visible during the work with the corresponding task), and records each interaction step conducted by the user. Furthermore, it controls the time of system usage: For each task there is a special amount of time available. The interaction time is started once the user has confirmed by pressing a button that he has understood the task and is ready to start. After the time available has run down (the remaining time is always displayed for the user) the systems' interfaces are locked and hidden so that no more interaction is possible and no more information can be derived from the graphical presentation. Figure 9-2 shows the interfaces of the document list and the document map system as used for the experiments.

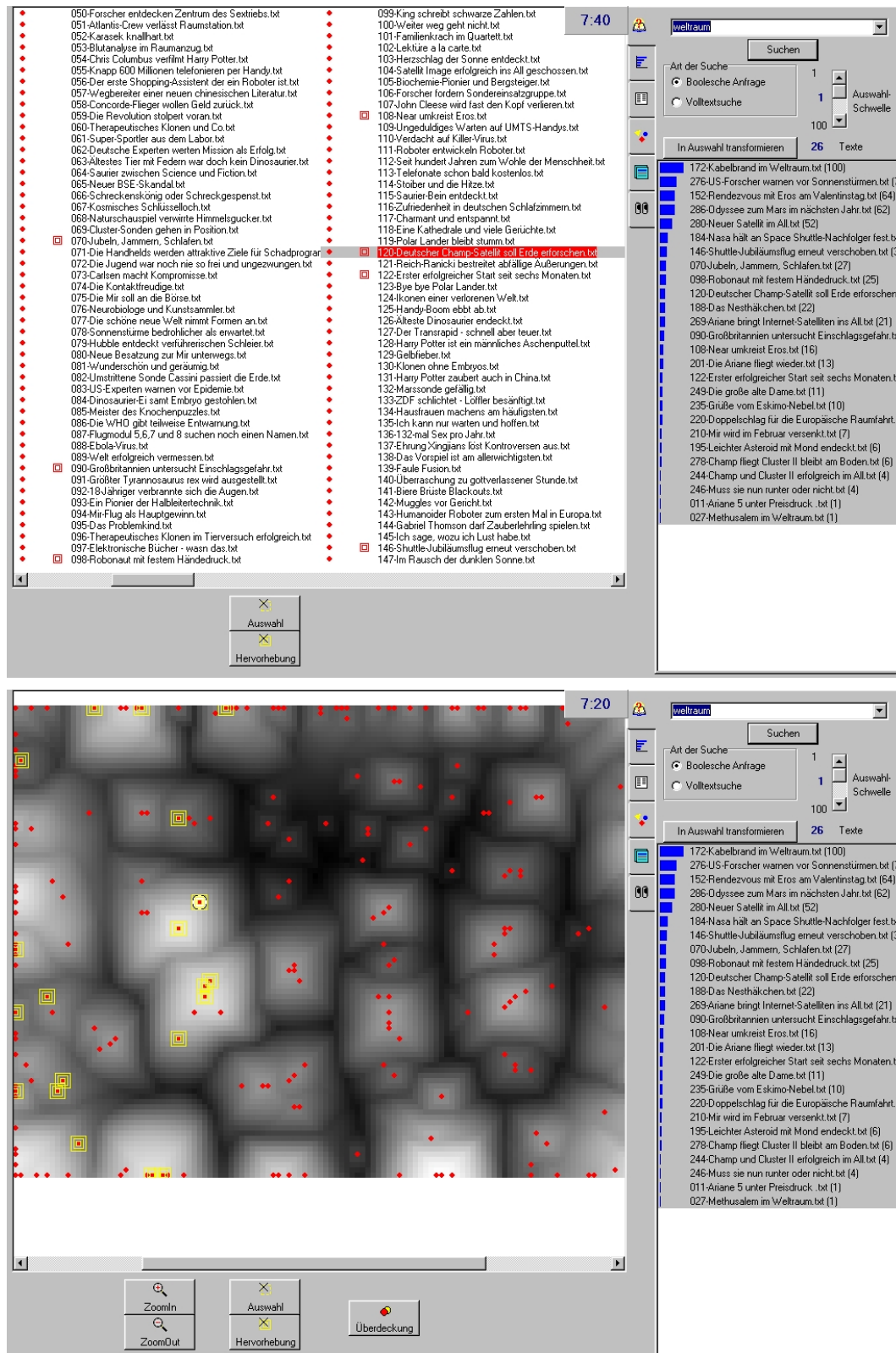


Figure 9-2: Screenshots of systems used in the comparative study. Only the visual workspace differs between list system (top) and map system (bottom). Besides, the map system includes a zoom function and an option to highlight document symbols where multiple texts are assigned to. The remaining time for dealing with the current task is displayed at the top right corner of the respective workspace.

9.2.2 Type of Measurement

In order to compare the effects of the two systems on the quality of task solutions achieved by their users, test subjects are randomly assigned to two groups, A and B. Both groups have to solve four different and typical corpus analysis tasks in succession. We use a 2x2-crossover design where each of the two test groups works with both systems: Group A works with the document map for tasks 1 and 2 and then with the document list for task 3 and 4, and vice versa for group B (cf. figure 9-3).

In general, the advantage of such a crossover design is that subjects become their own controls which reduces the error variance: Actually, group membership is the independent variable of the experiment (cf. section 9.2.5). Since we intend to measure the effects caused by the different systems, individual differences in the capability of test subjects (which may superimpose the treatment actually considered) have to be controlled. Though usually capability differences can be expected to be normally distributed within a sufficiently large sample, in practice this assumption may be violated more or less. The crossover design allows to consider a certain overall tendency of the experimental results. In particular, if *similar* tasks are performed consecutively by the two groups with a system change in between, the effect on the dependent variables should be reversed. This would give additional confidence in the experiment's results because influence caused by inter-individual differences are lowered. Furthermore, after the experiment test persons can give an overall assessment of the systems' suitability based at least on an indirect comparison.

A disadvantage of this design is, however, that the observations may be biased by so-called carryover effects, i.e. the persistence of certain treatment effects (e.g. acquired knowledge about the text corpus) from one task to the next. It is thus important to design the tasks to be independent from each other (cf. sections 9.2.5.5 and 9.2.6.6). But anyway, this is also true for continuous trials where subjects remain on the same treatment from the start of the experiment to the end if the single tasks shall be considered separately.

9.2.3 Cornerstones of a Test Design Inspired by Practical Scenarios

The possibility to generalize achieved experimental results depends heavily on the test scenario's closeness to reality. In this regard, crucial and important cornerstones of the comparative study are test subjects recruited, text collection used, and tasks assigned. Since the domain we are dealing with is concerned with corpus analysis in knowledge management – which involves advanced methods for accessing document collections – test subjects must have some professional skills. Thus we recruited computer science students from a graduate-

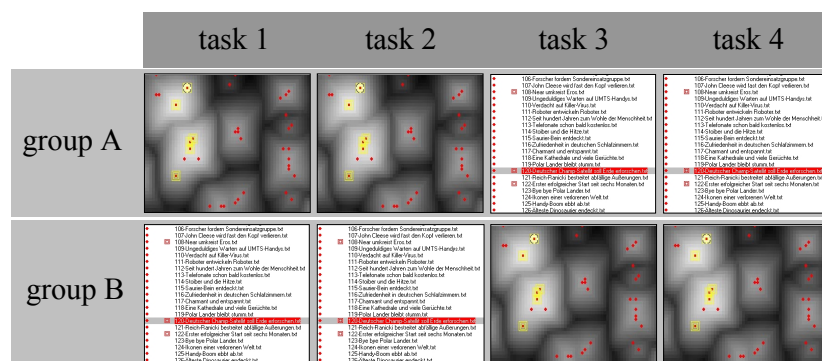


Figure 9-3: 2x2-crossover design with 4 tasks

level lecture ‘Introduction to Databases’, held in the winter term of 2000/2001 at RWTH Aachen. Altogether, 29 students were qualified to participate in the study (cf. section 9.2.5.2). From these test persons we could expect a sufficient basic understanding of working with and searching in document collections. Besides, we gave a special lecture about strategies and techniques for corpus analysis to ensure a common level of knowledge. Consequently, the students can be regarded as sufficiently skilled in document analysis tasks.

Closely connected to the choice of test subjects is the selection of a suitable document collection to be analyzed. Technical or scientific text corpora – as they are important in many real-world applications, cf. chapter 8 – require a deep understanding to adequately deal with them. Unfortunately, this cannot be expected from test subjects which have to cope with a difficult and unusual test situation anyway. Thus, a commonly understandable and interesting collection was chosen which has, moreover, some important properties that are shared by technical and scientific collections – as they occur in the application domain under consideration – as well: It was designed to be specialized in contents, it is homogeneous in style, comprises topics of different sizes and various degrees of relationship. Additional requirements met by the collection are discussed in sections 9.2.5.5 and 9.2.6.1.

The most important factor for assessing the task-adequacy of document maps is, of course, the selection of the tasks themselves. The tasks were designed with special care so that (a) they can be regarded as relevant in practice and (b) there is an objective reference solution which can be used to assess the quality of solutions achieved by the test persons. In order to ensure practical relevance, tasks were designed based on the task model for corpus analysis in knowledge management (chapter 2) and on results of an industry survey. Details on task definition are presented in section 9.2.6. Finally, the overall analysis situation simulated in the controlled experiment should be plausible and meaningful so that test subjects understand the purpose of their actions as such and are kept motivated.

9.2.4 Defining Measures for Assessing a System’s Task-Adequacy

The goal of the study is to compare the adequacy of two systems for a set of assigned tasks. How can the abstract idea of ‘task-adequacy’ be measured? In this section a set of measures is defined by applying the FCM⁵-method (according to [Fen92]) which helps to develop suitable and valid measures for complex factors in several steps: First, factors to be measured are subdivided into different criteria. For each criterion suitable metrics are defined in a second step. Having done this, the resulting measures can be tailored for each task considered.

Figure 9-4 presents an FCM-model for the factor ‘task-adequacy’: A system can be considered as adequate for performing a certain task if its solution can be achieved effectively and if the user is satisfied by the system support. Of course, one important aspect of ‘effectiveness’ is the quality of task solutions. Since in practice time always runs short quality alone is not relevant. Rather, quality and time necessary to complete a task have to be well-balanced. Thus, time, quality and user satisfaction are the criteria to be considered.

In order to determine the time needed to complete a task with the help of a system the time of system usage can be measured. User satisfaction can be assessed by a user ratings on preference scales. Measures for quality depend on the nature of the task: In the controlled experiment the solutions achieved by the test subjects are compared against an a priori defined reference solution. If certain items (e.g. documents) have to be selected by test subjects quality

⁵ Factor-Criteria-Metrics

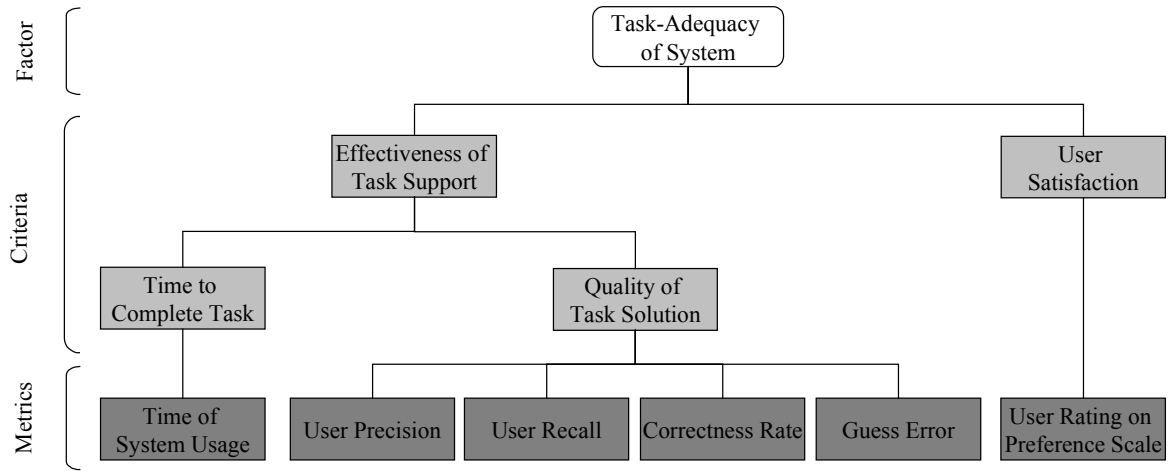


Figure 9-4: FCM-model for assessing a system's task-adequacy

can be assessed by the precision and recall of the solution achieved by the user. If questions have to be answered the fraction of correct answers matters. Finally, if quantities have to be guessed by the user the guess error is a good measure. The next sections present and discuss these measures in detail.

9.2.4.1 Quality of Task Solution

Let F be the set of items (e.g. documents or topics) found by a test subject and R denote the set of relevant items regarding the considered task. Then, the *user precision* p is defined as

$$p =_{\text{def}} \frac{|F \cap R|}{|F|}. \quad (9-1)$$

The *user recall* r is given by

$$r =_{\text{def}} \frac{|F \cap R|}{|R|}. \quad (9-2)$$

If necessary, precision and recall can be averaged for tasks where multiple sets of items (found and relevant) are considered.

Given guessed numerical values s_i for each relevant and found item and intervals $[l_i, u_i]$ of correct numerical solutions, the *average guess error* g is defined as

$$g =_{\text{def}} \frac{\sum_{s_i} f(s_i)}{|F \cap R|} \quad (9-3)$$

where $f(s_i) =_{\text{def}} \begin{cases} 0 & \text{if } s_i \in [l_i, u_i] \\ \min(|l_i - s_i|, |u_i - s_i|) & \text{otherwise} \end{cases}$

Finally, for a number c of correctly answered questions and a number q of questions asked, the *correctness rate* e is defined as

$$e =_{\text{def}} \frac{c}{q}. \quad (9-4)$$

9.2.4.2 Time to Complete Task

The ratio of quality and time is a good measure of effectiveness. Measuring both quantities at the same time, however, raises some problems in practice. First, test persons must understand that they have to optimize the ratio of quality and time. This is difficult if the operational target that should be reached cannot be defined clearly: Users might not be sure enough that the quality standard reached so far is sufficient to quit the task and still to achieve good results. Furthermore, individuals may cope differently with both degrees of freedom: Some users may stress quality more than time and others vice versa, independent from explicitly given orders. Though these effects can be expected to be observed equally within large samples, such individual differences are just another source of potentially disturbing influence in relatively small test samples. Apart from that, providing a completely flexible timeframe is impractical for organizational reasons.

Consequently, in this study the timeframe available is fixed. This does not produce any principal problems since it is not the absolute degree of quality of task solutions that is of interest but a comparison of results achieved by different groups. Thus, time is regarded as a standardizing quantity for the measures. However, if too much time is available differences of test persons who completed the task in different speeds and achieved similar quality cannot be assessed. Thus, a timeframe has to be chosen that induces a high time pressure: A perfect solution produced by a skilled system user should require slightly more time than available. At the same time we have to be sure that test persons will be able to complete a sufficient amount of work for comparing the quality of different results. Good timeframes can be found by self-testing and considering the time required by different test subjects in a trial experiment.

9.2.4.3 User Satisfaction

Qualitative questionnaires are applied to assess user satisfaction by ratings of test persons on a preference scale. More precisely, Likert scales (rating-scales for measuring attitudes) are used [Lik32]. Likert scaling is a one-dimensional scaling method. The ordinal scale measures the extent to which a person agrees or disagrees with a concept. Most commonly, 5 items are used which have, in principle, the following semantics: strongly unfavorable to the concept, somewhat unfavorable to the concept, undecided, somewhat favorable to the concept, strongly favorable to the concept.

After each task the test subjects are asked to rate the following concepts on a Likert scale with 5 items (cf. appendix B): System suitability for the assigned task (very poor – very good), assessment of effort for solving the task (very strenuous – very easy), and assessment of quality of own results (very poor – very good). The latter scale can be used to define a transparency measure: Can the user correctly assess his or her achieved quality? Let $Q^* \in [0,1]$ be an objective quality measure where 0 denotes the poorest and 1 the highest quality achievable. Furthermore, assume a numerical Likert scale encoding where 1 = very poor, ..., 5 = very good, and a correspondingly encoded quality scale where $Q = 1 \Leftrightarrow Q^* \in [0.0,0.2[$, $Q = 2 \Leftrightarrow Q^* \in [0.2,0.4[$, ..., $Q = 5 \Leftrightarrow Q^* \in [0.8,1.0]$. Then, for a given user rating $U \in \{1, \dots, 5\}$ the *transparency error* is defined by the difference $Q - U \in [-4,4]$ where 0 is the optimal transparency error and negative (positive) values indicate that the test subject assesses his or her quality to be better (worse) than objectively measured.

In a final questionnaire test persons are asked to rate their overall pleasure of using the two systems on a Likert scale: “How did working with the document map system (the document list system, respectively) appeal to you?” (very poorly – very well). Furthermore, the test subjects are asked to nominate their favorite system for solving each assigned task.

9.2.5 Variables and Influence Factors

Having defined comparative systems and measures for the experiment it is now important to summarize the variables used and to examine possible influence factors which may disturb them.

9.2.5.1 Independent and Dependent Variables

The *independent variable* in this experiment is group membership. Since each group is provided with a different system, being a member of a particular group implies the major influence on the results. The *dependent variables* are quality of task solution and user satisfaction. These variables are measured in principle as defined above. Yet, the measures still have to be adjusted to the concrete tasks assigned to the test subjects. The time to complete each task is fixed and thus implicitly contained in the quality measures so that actually the effectiveness of task support is measured.

It is important to examine factors that have an undesired influence on the experimental results (i.e. in particular on the dependent variables). Such *influence factors* have to be ruled out or at least controlled by the experimental design. The following sections present influence factors identified for this study and their handling in the experimental design.

9.2.5.2 Influence Factors Caused by Test Subjects

The major influence factors caused by test subjects are language skills, skills in using graphical user interfaces, and experience and background knowledge regarding the tasks to be tested. To eliminate such factors a pre-testing questionnaire filled out by potential test persons was evaluated. In detail, the following capabilities were checked:

- *Language skills*: Test persons were recruited from the lecture ‘Introduction to Databases’, held in the winter term of 2000/2001 at RWTH Aachen. Though the lecture is part of a Master program for international students and thus given in English, most of the participants are native speakers of German. Since a high variation of language skills would have been a serious influence factor we decided to conduct the comparative study in German. Due to the complexity of the text collection and tasks a very high ability to understand German is necessary. Test subjects whose mother tongue was not German were asked to rate their language capabilities on a 7-item scale. Only persons who judged their ability to understand German as being very good or excellent were accepted for participating in the experiment.
- *Basic experience in using a graphical user interface* had to be ensured. Only test persons with sufficient experience could participate.
- *Task-related background knowledge*: There was no special previous knowledge necessary for participating in the experiment. However, significant differences in experience with text retrieval systems and browsing might have influenced the results. Thus, a couple of questions in the pre-testing questionnaire was dedicated to matters of retrieval competence. This data was set in correlation to the achieved results to ensure that no significant influences occurred.
- *Collection-related background knowledge*: The collection used for this experiment consists of articles of an online news magazine. Because the time period covered by the selected texts is large and the fields of interests were carefully chosen (cf. section 9.2.6.1) it is unlikely that test persons are too familiar with its contents. But still we

asked for a list of online magazines read periodically by the test persons in order to correlate this fact with the results achieved, if necessary.

Test subjects that passed the pre-testing were randomly allocated to two equally-sized groups by drawing lots in order to avoid a correlation between their individual capabilities and group membership. To increase the motivation of the subjects, course credits could be earned and attractive prizes were provided as extrinsic motivation. The prizes were awarded independent of group membership so that being a member of a certain group was no advantage. Apart from that we tried to design the test scenario (text collection and tasks) to be interesting (intrinsic motivation).

9.2.5.3 Influence Factors Related to System Usability

Usability problems are a major influence factor caused by the systems. This factor is minimized by the following steps: First, the prototype system from which the experimental systems have been derived was tested regarding well-accepted criteria for usability (cf. chapter 7.5). The experimental systems were simplified by reducing their functionality (cf. section 9.2.1) in order to prevent the test subjects from being overwhelmed. A preceding cognitive walk-through ensured that each task can be solved well with both systems by using the remaining functions. Furthermore, a test study preceding the experimental session was conducted in order to detect relevant usability drawbacks. Finally, the test subjects passed a theoretical training of system features and active practice prior to the experiment. Since both systems differ only in their visual workspace possibly remaining usability problems cannot influence the results of a particular group.

9.2.5.4 Influence Factors Concerning Working Conditions

It is important that working conditions are the same for all test subjects. The experiment took place in a student computer room with 20 workstations. It was conducted in two directly successive sessions in each of which test persons were divided into the two test groups. Supervisors and the seating plan minimized the possibility that the subjects could copy the results of each other. Due to varying hardware available, the relative time-consuming function ‘term statistics’ (cf. section 9.2.1) was restricted for all users (by allowing to compute the statistics only for a limited number of documents) so that differences in CPU-speed did not matter for the interaction. The screen quality was comparable for all workstations.

9.2.5.5 Influence Factors Related to Document Collection and Tasks

The final group of influence factors is concerned with aspects of the document collection and the tasks actually assigned. To prevent influence towards the expected outcome of the experiments the following precautions were observed:

- *Objective reference solutions*: As a general requirement for a suitable text collection it is imperative to be able to define an objective reference solution for each task. The text collection used for the experiment provides meta-information which supports the definition of objective solutions (cf. section 9.2.6.1). The reference solutions of the tasks were worked out based on meta-information provided by the document source and a manual text analysis.
- *Compilation of text corpus*: The text collection was compiled independent of generating the document map using a pre-defined criteria guide. Corpus compilation and map generation were conducted by two separate scientists. Furthermore, the map was generated only when the tasks and their reference solutions were completely defined.

- *Document map generation:* In chapter 10 a semantic refinement method for document maps will be introduced that allows to define a bias towards an individual interest (see also chapter 5.8.4). For generating the document map this module (cf. chapter 10) was not used since such a bias would be a serious source of influence. Thesauri and stop word lists have been defined during the manual analysis of the document collection.
- *Mutual influence of tasks:* Since the outcomes of the tasks are evaluated separately it is important to avoid mutual influences. In particular, the achieved quality of solution for one task must not influence the solution quality of another task. An optimal procedure would be to test the tasks of the experiment with some distance in time (let's say one week) so that test subjects forget possibly helpful knowledge about the text collection they acquired, or to use a different collection for each task. Both solutions are not feasible due to organizational restrictions. Thus, possible influences were lowered by defining a suitable order of tasks and by considering different aspects of the text corpus for each task. A detailed discussion will be given in section 9.2.6.6.
- *Training lecture:* To ensure that all subjects had similar knowledge in analysis and retrieval tasks we performed a lecture on these matters prior to the experiment. This is particularly necessary because the domain of corpus analysis requires some skills which cannot be taken for granted. In particular, the usage of the systems and their features was presented (cf. 9.2.5.3) and general strategies for solving analysis tasks similar to the tasks given in the experiment have been taught. To avoid a bias towards a specific system the varying usage aspects of the systems (mainly related to the different workspaces) were treated equally in time and effort.

9.2.6 Designing Practically Relevant Tasks

Designing tasks for an empirical investigation is a crucial cornerstone: In order to gain observations which are practically significant, tasks have to be defined that meet relevance requirements of the application domain under consideration (cf. [Stü95]). In chapter 2 a task model for corpus analysis in knowledge management is presented which provides a structured taxonomy for characterizing typical analysis tasks. A survey in knowledge-intensive industries (cf. chapter 2.5) yielded a collection of practically relevant analysis jobs which were used as a rich source for defining the following set of test tasks. Section 9.2.6.6 summarizes the tasks discussed in the following sections and shows that many important characteristics of analysis jobs are covered by the defined task selection.

The way of presenting the set of tasks in the following is guided by the procedure applied for defining it: First, practically relevant aspects of analysis tasks are presented which lead to the formulation of a task that is ideal in the sense that it comes close to practical descriptions of analysis jobs. Such tasks occur in similar forms in different sectors and thus are worth to be examined in this investigation. In the second step the ideal task is altered with respect to important operational restrictions so that, on the one hand, the character of the ideal task is preserved as good as possible and, on the other hand, the task is approachable for an objective assessment. Criteria for operational tasks are

- restrictions in time, mental effort and task complexity for the test persons,
- the existence of an objective reference solution and the possibility to objectively assess the results achieved by the test subjects,

- a good rationale for each task so that the test subjects understand its value and are motivated to solve it,
- a possible support by both systems.

Finally, the tasks are adjusted to the used test collection, i.e. they are formulated accordingly and concrete answer sheets are defined. Furthermore, measures for assessing the quality of solutions are determined. The remainder of this section presents the test collection used for the tasks, develops tasks and sketches their reference solutions. Note that the task formulations presented in this section are condensed for a compact presentation. The original (German) task and answer sheets with additional explanations and comments can be found in appendix B.

9.2.6.1 Selection of a Suitable Text Collection

A suitable text collection for a controlled experiment has to meet some important requirements: First, it has to provide adequate meta-information for defining an objective reference solution for the tasks (cf. section 9.2.5.5). Second, its structure should be comparable to that of specialized collections (cf. chapter 8). In particular, the corpus should be restricted in content, different topics with various sizes and documents with different degrees of relationship should exist. Third, the corpus should consist of a non-trivial number of documents with a realistic text length. At the same time a limitation of the test subjects' mental effort has to be considered. Last but not least, the corpus should be interesting and easily understandable for the test subjects.

The text collection used for the experiments consists of 300 articles from the German online news magazine "Spiegel online" (www.spiegel.de/archiv) which appeared between July 1999 and October 2000. The intelligible articles with an average length of 342 words are from the categories 'Internet', 'culture' and 'science' (a detailed breakdown can be found in table 9-2, p. 172). Topics covering daily politics were excluded, and completely different subject areas have been inserted into the corpus in order to reduce the likelihood that test persons are too familiar with the collection's topics. The titles of the articles are used as document titles to be displayed by the systems. Each document has a randomly assigned number with which it can be identified (these numbers are used by the test persons to note their task results). Further-

SPIEGEL ONLINE - 05. Oktober 1999, 15:01, header (removed in experimental collection)
 URL: <http://www.spiegel.de/wissenschaft/0,1518,45293,00.html>

Raumfahrt category (removed in experimental collection)
Ariane 5 unter Preisdruck

150 Millionen Mark kostet zurzeit der Bau der europäischen Ariane 5. Die Amerikaner bauen derweil an einer eigenen unbemannten Rakete und werden den Satellitentransport voraussichtlich in zwei Jahren um 40 Prozent günstiger anbieten können. Wollen die Europäer den Wettlauf ins All nicht verlieren, müssen sie die Kosten senken. [...]

© SPIEGEL ONLINE 40/1999 links to related documents (removed in experimental collection)

Kontext:

- Raumfahrt: Stau vorm All (<http://www.spiegel.de/spiegel/0,1518,28183,00.html>)
- Raumfahrt: Pannenserie bei US-Raketenstarts (<http://www.spiegel.de/spiegel/0,1518,23655,00.html>)
- Verspäteter Start: Ariane ins All aufgebrochen (<http://www.spiegel.de/wissenschaft/0,1518,40146,00.html>)

Figure 9-5: Sample text (extract) from the online news magazine

more, titles will be displayed in the order defined by the numbers to prevent cluster artifacts which result from the genre-dependent, unusually descriptive titles.

Besides being interesting and understandable the choice of the collection brings along some other important features: A coarse category is assigned to each article by the editors of the online magazine and links to topically related texts (background information, similar and preceding articles) are given. This context information was used to compile the experimental collection and thus to get documents with different, objectively defined ‘degrees’ of relationship. The documents of the collection can be assigned rather clearly to a certain topic. The collection was designed to contain different subject areas with different sizes. Some topics can be combined to a more general super-topic, others are rather isolated.

Context data and categories provide objective information about related articles which was used to define the optimal results for each task. Of course, this meta-information was removed in the actual collection made available to the test persons (cf. figure 9-5). In summary, the requirements as discussed above are met by the chosen collection.

9.2.6.2 Task 1: Identify Topics of the Collection

For most analysis tasks the topical structure of the text collection under consideration is important. The analyst has to understand which topics are covered by the collection, which documents are related regarding their contents, how many texts are available for each topic, and how topics are related. For getting a first impression of a collection’s structure an understanding on a general rather than a detailed level is sufficient. Consequently, this task is concerned with recognizing topics, understanding the assignment of documents to topics, and grasping relationships among topics.

A general and ideal formulation of a task would be as follows:

- (IT1) *Determine the subject structure of the collection: Identify groups of topically related documents, name subjects or subject areas these groups deal with, and assess the approximate number of documents of each relevant topic.*

However, in this form the task is too complex and requires a subjective assessment and naming of topics. Achieved results would be difficult to evaluate and would hardly be comparable. In order to avoid this problem a list of possible topics, i.e. matching as well as non-matching subject titles, can be given. Note, however, that presenting this list *during* the working time would change the character of this task: Rather than exploring the collection and working out its structure it is likely that the test subjects just check the proposed topics regarding their occurrence in the collection.

Therefore, the test subjects were asked to work out the collection’s topics without being provided with the list of possible topics and urged to make notes. Only after the working time made available a list of occurring as well as non-occurring topics was presented. At this time the respective system was locked and invisible. The operational task was formulated in two parts which were presented in parallel:

- (T1a) *Find out topics for which texts can be found in the collection. If you find rather broad topics try to identify meaningful sub-topics. Afterwards you will receive a list with proposed topics and you will be asked to cross-mark those topics for which texts can be found in the collection.*

- (T1b) *Get an idea of the approximate number of documents for each topic. Afterwards you will be asked to guess the number of the proposed topics as precisely as possible.*

Table 9-1: Structure of answer sheet for task ‘identify topics’

topic	texts found?		number of texts	
	yes	no	#	don't know
space shuttles	yes	no	#	?
black holes	yes	no	#	?
....	yes	no	#	?

The definition of the term ‘topic’ was given on the task sheet: A ‘topic’ in the sense of this task is a higher subject with which at least two documents deal. The test persons were urged not to deal with single texts but to get an overview of topics and dominant sub-topics. Due to the nature of the text collection we could expect that the desired focus became clear quickly – an assessment that was confirmed by a preceding test study.

The structure of the answer sheet is depicted in table 9-1. The two parts of this task were evaluated separately. The proposed but non-occurring topics were carefully chosen so that clearly no document from the collection could reasonably be assigned to them. In total, 58 topics were proposed out of which 26 indeed occurred in the collection. The reference solution for the second part of this task takes into account that documents may be meaningfully assigned to different topics. All plausible assignments to the given list of topics were worked out manually, also considering the context information given by the editors of the online magazine. Consequently, the correct number of texts for each occurring topic is determined by an interval which is defined by the minimal and maximal number of texts which could be assigned to each topic. Table 9-2 presents the topical structure of the text collection used.

The quality of solution of the first part of the task is assessed by the achieved user precision p according to equation (10-16) and user recall r according to equation (9-2) where F is the set of topics found by the test subject and R is the set of occurring topics. Regarding the second part, the guess error g as defined by equation (9-3) is determined where F and R are defined

Table 9-2: Topical analysis of the test collection: minimal / maximal number of texts assignable to each topic.

topic	max. # texts	min. # texts	topic	max. # texts	min. # texts
space shuttles	36	27	brain research	11	5
space probes / satellites	35	27	computer viruses	8	7
biological viruses	28	24	handheld / PDAs	8	6
sun and eclipse of the sun	24	24	book review TV show	8	8
space station	23	13	cloning	7	7
Harry Potter	21	21	Stephen King	6	3
Noble prize	17	15	magnetic levitation train	6	6
mammoths / dinosaurs	15	15	Oktoberfest	5	5
sexuality	15	15	research politics	5	5
cell phones / UMTS	14	13	software patents	4	4
robots / RoboCup	13	12	human ancestors	4	4
electronic books	12	9	BSE / mad cow disease	4	3
asteroids and meteoroids	12	10	outlier texts	4	4
space telescope Hubble	11	8	Σ	356	300

as before and the s_i are the guess quantities for correctly identified topics as given by the test subject. The respective intervals $[l_i, u_i]$ are defined by the minimal and maximal number of texts which could be assigned to each topic (cf. table 9-2).

9.2.6.3 Task 2: Find Outliers in the Collection

The observance of quality criteria for documents and text collections is an important task in many areas of working with documented knowledge. One criterion is to avoid ‘outliers’ in text collections. This is a crucial point for many applications, e.g. for identifying misplaced texts in documentation systems or testing the homogeneity of text groups (for instance in order to define categories). This kind of task is also discussed in the case study about product documentation where a basic quality criterion is that sections of user manuals must not be semantically isolated (cf. ‘checking the quality of user documentation’ in section 8.2.1). Ideally, a task to be tested could look as follows:

(IT2) Check if the given document groups are semantically homogeneous.

Similar to the first task, there would be too much space for subjective interpretations. Furthermore, texts can be examined on different levels of abstraction. For the defined operational task clear ‘inconsistencies’ have to be constructed. Thus, documents which undoubtedly do not belong to any other topic covered by the collection were added to the text database. Accordingly, the operational task was formulated as follows:

(T2) Find topical outliers, i.e. single texts which do not belong to any other topic of the collection.

As an additional aid for the test subjects the number of outliers in the collection was given. The four correct outlier documents which were added to the database deal with danger of tidal waves, reasons for bad breath, the German singer for the European Song Contest 2000, and the cause of death of Ludwig van Beethoven (cf. topics in table 9-2). The quality of the task’s solution is assessed by the achieved user precision p according to equation (10-16) and user recall r according to equation (9-2) where F is the set of documents found by the test subject and R is the set of real outlier documents.

9.2.6.4 Task 3: Find Classification Errors

Since many companies use classification systems to organize their text material (e.g. in catalogues or file systems) the correct classification of texts plays an important role. In practice, tasks like checking if the available categories reflect the content-structure of the corpus or testing if documents are assigned to correct classes are relevant. This task shall help to assess in how far the test subjects are able to identify clearly misclassified documents:

(T3) Find documents which are assigned to a wrong class. Identify only clear mistakes. Do not search for texts which could also be assigned to an alternative class.

To test this all documents in the workspace of the respective system were color-coded regarding the following classes:

- | | |
|--------------------------------|---------------------------|
| • viruses and diseases | • literature |
| • technology and communication | • research |
| • leisure and amusement | • means of transportation |
| • patents | • space and sun |

Note that the outliers from task 2 were color-coded separately and were explicitly excluded from this task. The classification errors constructed for this task were designed in a way that a

Table 9-3: Misclassified documents for task 3

title	topic	correct class	assigned class
Ein harter Brocken (A Tough Piece)	new Harry Potter novel	literature	research
Jubeln, Jammern, Schlafen (Cheering, Moaning, Sleeping)	launch of Sojus capsule	space and sun	leisure and amusement
Schlimmer als Melissa (Worse than Melissa)	computer virus	technology and communication	space and sun
Luzia sorgt für Aufsehen (Luzia Attracts Attention)	anthropology	research	literature

wrong classification is obvious. Altogether, four misclassified texts could be found in the test collection (cf. table 9-3). The quality of task solutions can again be assessed by the achieved user precision p according to equation (10-16) and user recall r according to equation (9-2). In this case, F contains all documents identified by the test subject for this task, and R is the set of real misclassified documents.

9.2.6.5 Task 4: Understand Context and Relate Documents

Working with documented knowledge involves tasks such as understanding the topical context of single texts and grouping related texts. The special process of associating documents and condensing their contents is important in various areas like generating product documentation (cf. section 8.2.1), synchronizing brain storming notes, or writing newspaper articles. A corresponding task would be:

(IT4) *Work out topically strongly related groups of texts and summarize their contents.*

Like before, the degree of subjectivity would be too high to objectively assess the quality of solutions in a controlled experiment. In order to define an operational task only a subset of all documents was considered and a priori given. The task was split up in two sub-tasks:

(T4a) *Consider only the highlighted subset of documents. Work out small groups of topically strongly related texts with not less than 2 and not more than 5 documents. Each text may belong to only one group. An reference solution consists of 11 groups.*

(T4b) *Get an overview of the main subjects of the groups you identified. After working time you will receive a list of statements and you will be asked to decide whether each statement characterizes a group of related texts. Make notes for this purpose.*

For this task a subset of 32 texts dealing with space missions was selected and highlighted in the workspace of the respective tool. The test leader verbally pointed out to the test persons that the desired groups typically are series of articles about a particular subject. Due to the nature of the collection used for this experiment we could expect that closely related documents can be recognized intuitively. A preceding test study could indeed confirm that test persons find the desired relationships (independent of group membership). Furthermore, the clear description of group characteristics (2–5 texts and 11 groups) helps to understand the degree of granularity of the optimal solution.

An objective reference solution for T4a was defined based on the context information provided by the online magazine and a manual text analysis. Example topics of correct text groups are

- European carrier rocket ‘Ariane’,
- danger of asteroid impact on earth,
- space probe ‘Near’ circles round asteroid ‘Eros’, or
- scrapping of space station ‘Mir’.

For task T4b statements were worked out that either summarize correct groups or contain false statements. False statements either characterize groups that were not considered or contain clear mistakes that can be identified quickly by reading the title or the first sentences of the articles. However, these false statements are plausible, i.e. it is not possible to decide whether they are correct without knowing the main subjects of the considered articles. Examples of such statements are:

- “Space station ‘Mir’ is to be traded on stock exchange in order to prevent its scrapping.” (*correct*)
- “Space probe ‘Near’ circles round earth and will be in a good position to send pictures of asteroid ‘Eros’ to earth on Saint Valentine’s Day.” (*wrong*)

The quality of solutions for task T4a is measured by the averaged user precision p^* and the averaged user recall r^* . More precisely, let $R^* = \{R_1, \dots, R_{11}\}$ be the correct partition of the considered set of documents and $F^* = \{F_1, \dots, F_m\}$ be the partition constructed by the test person. The average user recall is then computed by

$$r^* = \frac{1}{|R^*|} \sum_{F_i} \max_{R_j} \left\{ \frac{|F_i \cap R_j|}{|R_j|} \right\}.$$

Analogous, the average user precision p^* is defined. The quality of solutions for task T4b is determined by the correctness rate e according to equation (9-4).

9.2.6.6 Summary, Classification and Mutual Influence of Tasks

Having described the tasks for the study in detail, table 9-4 summarizes the most important operational aspects. As discussed in section 9.2.4.2 the time available to solve the tasks (more precisely: the time for using the systems) was chosen to induce a high time pressure. Preceding trial experiments with different test persons yielded the timeframe given in the table. Table 9-5 presents the characterization of the proposed test tasks using the task model from chapter 2. It can be seen that most characteristics of typical document analysis jobs are covered. The important aspect of maintaining the category structure of document collections is addressed in task 3. An important remaining aspect is to discuss the possible mutual influence of tasks assigned to the test subjects.

Since the goal of this study is to separately evaluate the single hypotheses it is necessary to rule out mutual influences of the tasks. Of course there will be certain training and habituation effects while working with the systems and the collection in the context of the study (course of the study, motivation changes and the like) which cannot be prevented completely. Training effects regarding tasks and collections can be lowered if the single tasks are concerned with different aspects (i.e. different topical areas, different levels of granularity, etc.) of the collection. This section argues that possible mutual influences by learning effects are low and can thus be neglected. For each relevant combination of tasks i and j the following list makes plausible that there is no (significant) influence of task i on task j ($T_i \rightarrow T_j$):

Table 9-4: Operational aspects of defined tasks

task	T1	T2	T3	T4
short description	identify topics	find outliers	find classification errors	understand context and relate documents
primary characteristic of task	overview of topics, assignment of documents to topics, topical relationships	identification of unusual documents, more detailed understanding of text-topic relationships	understanding relationships between texts and external specifications	detailed insight into relationships among single documents
additional information	–	–	color-coding of documents regarding given classification	highlighting of documents considered in this task
output produced by test subjects	cross-marking of assumed topics in a list, approximate number of texts (after working time)	numbers of assumed outlier documents	numbers of assumed misclassified documents	table with numbers of documents for each assumed group, binary answers to questions about contents (after working time)
reference solution defined by	categories and context information given by online magazine	documents which are definitely off topic (taken from clearly different categories)	documents with altered classification (using simplified classes)	context information given by online magazine
measures for quality	user precision, user recall, guess error	user precision, user recall	user precision, user recall	user precision, user recall, correctness rate
time available	8 min. system usage + sufficient time to fill out answer sheet	5 min.	5 min.	10 min. system usage + sufficient time to fill out answer sheet

- (T1 → T2) Task 1 has to be performed under high pressure of time. It is thus unlikely that outliers which are relevant in task 2 (which requires a detailed analysis of the collection) are recognized already in task 1 (which requires a coarse analysis). The degree of knowledge of the collection achieved in task 1 might have an influence on the quality with which task 2 can be performed. However, the list of topics given after having performed task 1 contains all topics covered by the collection. No topic given in this list is the topic of an outlier document from task 2. Furthermore we can expect that the pure recognition of topics is done with similar quality by both groups.
- (T1 → T3) Recognizing classification errors is independent of the collection's overall topical structure since this task focuses on single classes. Furthermore, the color-coding of documents in the systems' workspaces indicates the overall class structure. Thus, differences in knowledge about the topical structure are compensated anyway.
- (T1 → T4) In contrast to task 1, task 4 requires a very fine-granular analysis of a particular sub-collection of documents. Due to the narrow time-restriction in task 1 it is highly unlikely that these detailed document relationships can be detected. Furthermore, the test persons are instructed to cope with a coarse overview.
- (T2 → T3) Outlier documents from task 2 are excluded and marked correspondingly in task 3. Furthermore, identifying classification errors is independent from knowing outliers.
- (T2 → T4) Task 4 is concerned with a special sub-collection. Knowing outliers is not relevant here.

Table 9-5: Characterization of test tasks using the task model (cf. chapter 2.6)

T1: identify topics						
T2: find outliers						
T3: find classifications errors						
T4: understand context and relate documents						
goal of interaction	making use of documents	learn	✓			✓
		condense documents	✓			
		select documents	✓			
	maintaining documents	control quality		✓	✓	
		assure quality				
dynamics of focus of interest	fixed		✓	✓	✓	
	adaptive					✓
resource considered	information from documents		✓	✓	✓	✓
	meta-information	document attributes		✓		
		structural information	✓		✓	✓
method of interaction	explorative		✓			✓
	goal-directed		✓	✓	✓	
granularity	overview					✓
	details		✓	✓	✓	
categories	relevant	use existing	not relevant	✓		✓
		use classes	check classes		✓	
			use classes			
			categorize			
focused relationship	external: document – specification			✓		
	inherent	document – document	✓			
		document – topic			✓	✓
		topic – topic				✓
mode of communication	recognize (reflect, physically passive)		✓	✓	✓	✓
	specify (physically active)		✓			

(T3 → T4) Knowing misclassified documents is not of help for task 4. Furthermore, all documents of the sub-collection dealt with in task 4 belong to the same class in task 3. The classification gives thus no hints regarding a certain grouping required in task 4.

Now the formal hypotheses can be formulated which are tested statistically in the empirical investigation.

9.2.7 Formal Hypotheses

In this section the hypotheses from section 9.1 are formulated technically. This is done in two steps: First, one-sided hypotheses are defined, differentiated by tasks and measures. These hypotheses include the direction of the expected effect of group membership on the achieved results. In contrast, the corresponding null hypotheses (which are actually tested statistically) are two-sided, i.e. they only state that there is no difference in means between the two groups. If a null hypothesis can be rejected due to the test result the corresponding alternative hypothesis can be accepted if the means of the considered distributions differ from each other in the assumed direction.

Each hypothesis from section 9.1 is subdivided into a set of formal hypotheses: For each relevant measure a separate hypothesis is formulated which allows a differentiated consideration of results. A group is said to perform better for a particular task (regarding effectiveness) than the comparative group if the group achieves significantly better results regarding at least one measure and there is no tendency against this group regarding the remaining relevant effectiveness measures.

The hypotheses are based on the measures and tasks defined in sections 9.2.4 and 9.2.6. Recall that group A uses the document map system for tasks 1 and 2 and the document list system for tasks 3 and 4, and group B uses the document list system for tasks 1 and 2 and the document map system for tasks 3 and 4.

9.2.7.1 One-Sided Alternative Hypotheses

Regarding effectiveness the following hypotheses are defined:

- (H1_{T1a,p}) Finding out topics of a collection (T1a) can be solved more precisely (regarding precision p) by group A than by group B.
- (H1_{T1a,r}) Finding out topics of a collection (T1a) can be solved more completely (regarding recall r) by group A than by group B.
- (H1_{T1b,g}) Assessing the approximate number of documents for topics (T1b) can be solved with a lower guess error g by group A than by group B.
- (H1_{T2,p}) Finding topical outliers in a collection (T2) can be solved more precisely (regarding precision p) by group A than by group B.
- (H1_{T2,r}) Finding topical outliers in a collection (T2) can be solved more completely (regarding recall r) by group A than by group B.
- (H1_{T3,p}) Identifying classification errors in a collection (T3) can be solved more precisely (regarding precision p) by group B than by group A.
- (H1_{T3,r}) Identifying classification errors in a collection (T3) can be solved more completely (regarding recall r) by group B than by group A.
- (H1_{T4a,p}) Associating related documents in a collection (T4a) can be solved more precisely (regarding average precision p^*) by group B than by group A.
- (H1_{T4a,r}) Associating related documents in a collection (T4a) can be solved more completely (regarding average recall r^*) by group B than by group A.
- (H1_{T4b,e}) Getting an overview of main subjects of associated groups (T4b) can be solved more correctly (regarding correctness rate e) by group B than by group A.

For tasks T1, T2, T3, and T4 and all quality measures the following hypothesis is defined:

- (H1_{Ti,transp}) Users of the document map can better assess the quality of their results (regarding transparency error $Q-U$) than users of the document list.

Regarding user satisfaction the following hypotheses are defined for tasks T1, T2, T3, and T4:

- (H1_{Ti,suit.}) Users of the document map judge their system to be more suitable for the assigned task than users of the document list.
- (H1_{Ti,effort}) Users of the document map judge the effort necessary for solving the assigned task to be lower than users of the document list.

And for task T1 we define additionally:

- (H1_{T1,fam.}) Users of the document map feel more familiar with the text corpus after having performed task T1 (getting an overview of the collection) than users of the document list.

Finally, for the overall preference judgment we define:

- (H1_{pref.}) Test subjects enjoy working with the document map more than working with the document list system.

9.2.7.2 Two-Sided Null Hypotheses

Correspondingly, the following null hypotheses are tested regarding effectiveness:

- (H0_{T1a,p}) Group membership does not influence the precision p for finding out topics of a collection (T1a).
- (H0_{T1a,r}) Group membership does not influence the recall r for finding out topics of a collection (T1a).
- (H0_{T1b,g}) Group membership does not influence the guess error g for assessing the approximate number of documents for topics (T1b).
- (H0_{T2,p}) Group membership does not influence the precision p for finding topical outliers in a collection (T2).
- (H0_{T2,r}) Group membership does not influence the recall r for finding topical outliers in a collection (T2).
- (H0_{T3,p}) Group membership does not influence the precision p for identifying classification errors in a collection (T3).
- (H0_{T3,r}) Group membership does not influence the recall r for identifying classification errors in a collection (T3).
- (H0_{T4a,p}) Group membership does not influence the average precision p^* for associating related documents in a collection (T4a).
- (H0_{T4a,r}) Group membership does not influence the average recall r^* for associating related documents in a collection (T4a).
- (H0_{T4b,e}) Group membership does not influence the correctness rate e for getting an overview of main subjects of associated groups (T4b).

For tasks T1, T2, T3, and T4 and all quality measures the following hypothesis is tested:

- (H0_{Ti,transp}) Group membership does not influence the transparency error $Q-U$ for self-assessing the quality of achieved results.

Regarding user satisfaction the following null hypotheses are tested for T1, T2, T3, and T4:

- (H0_{Ti,suit.}) Group membership does not influence the judgment regarding system suitability for the assigned task.
- (H0_{Ti,effort}) Group membership does not influence the judgment regarding effort necessary for solving the assigned task.

For task T1 we define in addition:

- (H0_{T1,fam.}) Group membership does not influence the judgment regarding familiarity with the text corpus after having performed task T1 (getting an overview of the collection).

For the overall preference judgment the following null hypothesis is tested:

- (H₀_{pref.}) Test subjects do not enjoy working with one of the systems more than working with the other.

9.3 Preparing and Conducting the Experiment

This section sketches how preparations for the experiment were made and describes the course of the experimental session in detail.

9.3.1 Pilot Study

Some weeks prior to conducting the experiment we performed a pilot study in order to test the design and to identify flaws and avoidable difficulties (the design presented in section 9.2 is that used in the final experiment). Therefore, eight test subjects (students of computer science and engineering who were not involved in this research project and did not participate in the final experiment) were recruited and passed a complete pilot session, including a theoretical and practical training and the trial experiment.

In particular, the following aspects were checked and, if necessary, improved: First, we made sure that the training did not include any bias towards or against any system and that it was sufficient to put test subjects in the position to perform the intended corpus analysis tasks with both systems. We checked that test persons correctly understand the tasks and cleared up remaining usability problems with the tools. Using the experimental results we could confirm by a correlation test that the tasks are indeed independent from each other. Remaining errors in the reference solutions and problems with the answer sheet for task T1a (table 9-1) were corrected (mainly ambiguities concerning the proposed topics). The original plan for task T4 was to explore the collection by starting with a given document and to construct a text grouping, the reference solution being based on the starting document's context information (cf. section 9.2.6.1). Though we defined clear topical constraints for this task there was still too much space for individual interpretations so that the target documents were given in the final version of this task (cf. section 9.2.6.5).

Also, some minor changes regarding the collection were necessary: Both test groups identified supposed outliers in the collection, the correctness of which could be argued. Such controversial documents were replaced. Finally, the pilot study yielded the timeframes for solving the tasks in the main experiment (table 9-4).

9.3.2 Generating the Document Map for the Experiment

The document map used for the pilot experiment was generated after the text collection and the tasks were completely defined. Due to minor changes in the text collection, as discussed above, a new map had to be generated for the final experiment. Like before, this was done only after the changes were completed. The used stop word list and thesaurus (containing synonyms and compound words) was generated during the manual analysis of the collection. For the purpose of a training session a second collection (consisting of a different set of documents from the same source) was compiled and a corresponding map was generated. Note that the semantic refinement module which will be presented in chapter 10 was not used (cf. section 9.2.5.5). Parameters of the maps are given in table 9-6. The map used in the experiment is printed in appendix D.

Table 9-6: Parameters used for generating the document maps

parameter	training	experiment
# documents	77	300
weighting scheme	TF	TF
similarity measure	cosine	cosine
dimension document space	60	150
average stress	0.047	0.088
size of SOM	100 × 80	120 × 90
training steps per document	25	30

9.3.3 Introductory Lecture and Practical Training

Corpus analysis tasks require a good understanding of strategies and problems of accessing text collections. Since the intended user of a corpus analysis system like that studied in this work is an experienced analyst, it is necessary to recruit test persons who come close to such experts as far as their skills are concerned. This was taken into account when we decided to ask computer science students at a graduate level – from whom we can expect a basic understanding of the field – to participate in this study. Additionally, a special lecture on corpus analysis and text retrieval was given which also introduced the two systems and their usage.

The lecture first provided some theoretical background on the basics of analyzing and accessing document collections. In particular, typical tasks which arise in practice were sketched in a scenario-based way and general query-driven and explorative access strategies were introduced and contrasted. Then, the two systems were presented, the semantics of the common functions were discussed, and their application to tasks similar to those which occurred in the experiment was worked out. Finally, different strategies to solve the tasks were demonstrated and discussed. A detailed presentation of the lecture (including the slides used) can be found in [Seel01].

Directly before the experiment, test persons could make themselves familiar with both tools in a free exercise. A small sample collection consisting of news articles from the same source as the texts used in the experiment was provided for this purpose. For this training session the test persons used a combined map and list tool, i.e. they could switch between both workspaces at will. Selections, symbols and highlighted documents were transferred from one workspace to the other, so that differences and common characteristics could be easily explored. There were no special working instructions, but we advised the students to carefully deal with both systems.

9.3.4 Course of the Experiment

The experiment was held in November 2000 in a student computer room at the Computer Science Department of the Technical University of Aachen. Altogether, 29 students participated in the experiment in two directly successive sessions. In each session the test subjects were divided into two test groups. The experimental sessions were attended by 4 supervisors who were instructed to control the technical course of the experiment, to give advice if usability questions arise, to answer questions regarding the general understanding of tasks (no recommendations for task solutions were allowed) and to record any important event. During the experiment the following data was collected:

- the solutions for each task and preference judgments (noted on the task sheets and questionnaires by the test subjects), and
- an automatically generated interaction protocol in which each action carried out with the tools along with all relevant parameters and time-stamps is recorded.

Test subjects were provided with a quick reference on how to use both text-access systems (cf. appendix C), as well as paper and pens for making notes. Table 9-7 summarizes the course of the experiment.

Table 9-7: Course of the experiment

Introductory Lecture (November 6, 2000)
<ul style="list-style-type: none">• Introduction to analyzing and accessing document collections.• Presentation of systems.• Test subjects fill in pre-testing questionnaires (personal data, previous knowledge, cf. section 9.2.5.2).
Experimental Session (November 10, 2000)
<ul style="list-style-type: none">• Randomizing sample: test subjects draw lots for group membership.• Introduction:<ul style="list-style-type: none">◦ Explaining course of session,◦ brief refresh of analysis scenarios◦ sketching of tasks to be solved.• Free exercise (15 minutes) with training collection, workspaces can be switched at will.• Systems are changed to the experimental modus (experiment control system, cf. section 9.2.1).• Handing out task sheets and questionnaires.• Performing task 1. Handing out additional answer sheet after working time. Filling in questionnaire for task 1.• Performing task 2. Filling in questionnaire for task 2.• Performing task 3. Filling in questionnaire for task 3.• Performing task 4. Handing out additional answer sheet after working time. Filling in questionnaire for task 4.• Handing out and filling in final questionnaire.

9.4 Results of the Experiment

This section starts with a specification of the statistical test procedures used for evaluating the formal hypotheses. After that, the two groups of test subjects are statistically characterized and compared. Then, results regarding quality of solutions for the different tasks and results of the qualitative questionnaires are presented. In the following we check correlations between the single dependent variables and list free comments on the systems as stated by the test persons. Finally, some results derived from the interaction protocols recorded by the systems are presented. The discussion and interpretation of the overall experiment will be given in section 9.5.

9.4.1 Statistical Test Procedures Used

This section briefly sketches the test procedures used for evaluating the hypotheses from section 9.2.7.2. An overview of methods of test statistics can be found in [Lore84]. Basically, we use Student's t -test to compare the means between two samples. More precisely, the two-sample, two-sided unpaired t -test is used to test the null hypothesis that the two population means corresponding to the two random samples are equal. However, it is important to be aware of the assumptions made by this test and the circumstances under which it is not valid: The t -test (and the corresponding tabulated critical values) requires two independent samples and assumes that within each sample the values are independent and identically normally distributed (i.e. same mean and variance). Fortunately, in practice the assumption about the normal distribution can be relaxed: According to [MRB89] “[...] *it can be shown that the distribution of the t statistic possesses nearly the same shape as the theoretical t distribution for populations that are nonnormal but possess a mound shaped probability distribution.*” Checking the frequency distribution of the results we observed showed that this assumption holds for the task quality data. If, however, the variances of the distributions are significantly different, Welch's T -test is used instead of Student's t . For checking the homogeneity of variances we use a variance quotient test (F -test) where the significance threshold for the error probability α is set to 0.2. In addition to Student's and Welch's test we use the Wilcoxon test for the task quality data as a further confirmation of the achieved results. The latter test does only require that both distributions possess the same form.

Using statistical trend tests for task solution quality measures is no problem since the data is naturally encoded by numbers. It is also possible to apply such trend tests to Likert scale questions (regarding user preference) once the ordinal scale items are encoded by numerical values (e.g. 1–5). Then, it is appropriate to determine means and standard deviations for the answers since the resulting numbers provide a feeling for the direction of the average answer. Again, for applying a t -test we have to make sure that the frequency distributions of results are at least mound shaped. Since the answer distributions for some of the questions are more or less far from being mound shaped (in particular, this is the case for some answers in the pre-testing questionnaire and the distribution of the transparency error as introduced in section 9.2.4.3) we use the simple χ^2 -test in such cases.

Finally, the procedure for testing the hypotheses is as follows: The two-sided null hypotheses are tested using an appropriate statistical trend test. If a null hypothesis has to be rejected on a certain level, the means of the samples are compared in order to decide whether the one-sided alternative hypothesis can be accepted. A test result is regarded as being statistically significant if the error probability α of the test does not exceed 0.05. If the error probability α does exceed 0.05 but is less than 0.1 we speak of a tendency.

9.4.2 Statistical Characterization of Samples

Prior to presenting the results of tasks and questionnaires a statistical characterization and comparison of the two randomized samples shall be discussed. Using a pre-testing questionnaire (cf. section 9.2.5.2) we collected data about important capabilities that may influence the experiment's results. Table 9-8 presents the averaged results. Regarding language skills, all members of group A were native speakers (we assumed the highest rating value for them), two members of group B were non-native speakers who self-assessed their language skills as being very good. Figure 9-6 visualizes the means of the technical skills for both groups. The ratings regarding document search experience and competence are rather high – an intended characteristic since the aim was to simulate the target group of analysis experts.

Table 9-8: Statistical characterization of samples: mean μ , standard deviation σ , $N_A = 15$, $N_B = 14$

feature	group A		group B	
	μ	σ	μ	σ
age	23.36	1.45	23.50	2.85
number semesters in informatics	7.07	2.79	7.63	5.19
language skills ¹⁾	7.00	0.00	6.81	0.54
experience Windows GUI ²⁾	3.43	0.65	3.38	0.50
frequency of using search engines and catalogues ³⁾	3.29	0.73	3.00	0.82
frequency of browsing ³⁾	3.64	0.50	3.75	0.45
frequency of searching in libraries ³⁾	2.14	0.53	2.06	0.25
self-assessment of Internet search-competence ²⁾	3.00	0.78	2.88	0.62
self-assessment of library search-competence ²⁾	2.21	0.80	2.63	0.62

rating scale values:
 1) 7 = excellent, ..., 1 = weak
 2) 4 = expert, 3 = experienced user, 2 = basic knowledge, 1 = no experience
 3) 4 = daily, 3 = weekly, 2 = sometimes, 1 = never

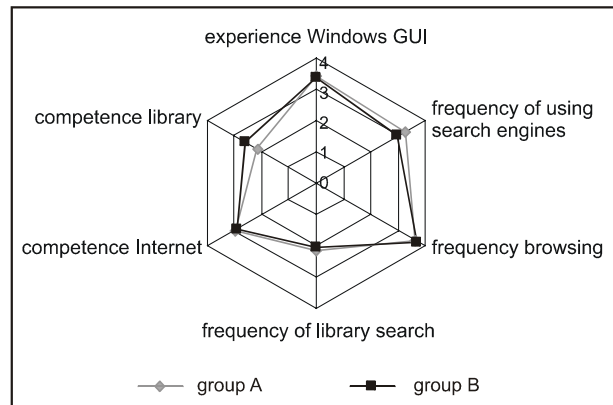


Figure 9-6: Capabilities of test subjects (means) regarding relevant background knowledge

Table 9-9: Statistical comparison of samples regarding significant differences

feature	age	semester	language skills	experience Windows GUI	frequency of using search eng.	frequency browsing	frequency of library search	competence Internet	competence library
χ^2 -test	0.722	0.276	0.971	0.811	0.651	0.990	0.456	0.777	0.040

Note that the technical skills are not independent from each other. Experience in using the Windows GUI is correlated to Internet search competence to a degree of 0.62 within the total sample, search engine experience and Internet search competence (0.6) and Windows GUI and browsing experience (0.52) are correlated rather clearly as well. Remaining correlation coefficients have an absolute value below 0.5. Finally, no test person periodically read the online magazine (or similar magazines) the document collection for the experiment was compiled from and could thus possess special knowledge about the text corpus.

Table 9-9 presents the results of the χ^2 -test for comparing the rating distributions of both groups. It turns out that the self-assessment in library search competence differs significantly between both groups ($\alpha = 0.04$). However, none of the dependent variables (task results and answers to questionnaires) is correlated to the rating value for library search competence with an absolute value of more than 0.48. Thus, we can neglect this difference between the groups.

Table 9-10: Experimental results: statistical characterization of solution quality for tasks
(mean μ , standard deviation σ , $N_A = 15$, $N_B = 14$)

task	measure	group A		group B	
		μ	σ	μ	σ
1a	precision	0.83	0.13	0.84	0.15
1a	recall	0.55	0.11	0.55	0.16
1b	guess error	3.44	2.16	5.53	2.71
2	precision	0.47	0.34	0.18	0.36
2	recall	0.38	0.31	0.11	0.22
3	precision	0.33	0.34	0.69	0.27
3	recall	0.17	0.12	0.84	0.31
4a	precision	0.50	0.22	0.76	0.17
4a	recall	0.36	0.20	0.41	0.21
4b	correctness rate	0.51	0.09	0.53	0.08

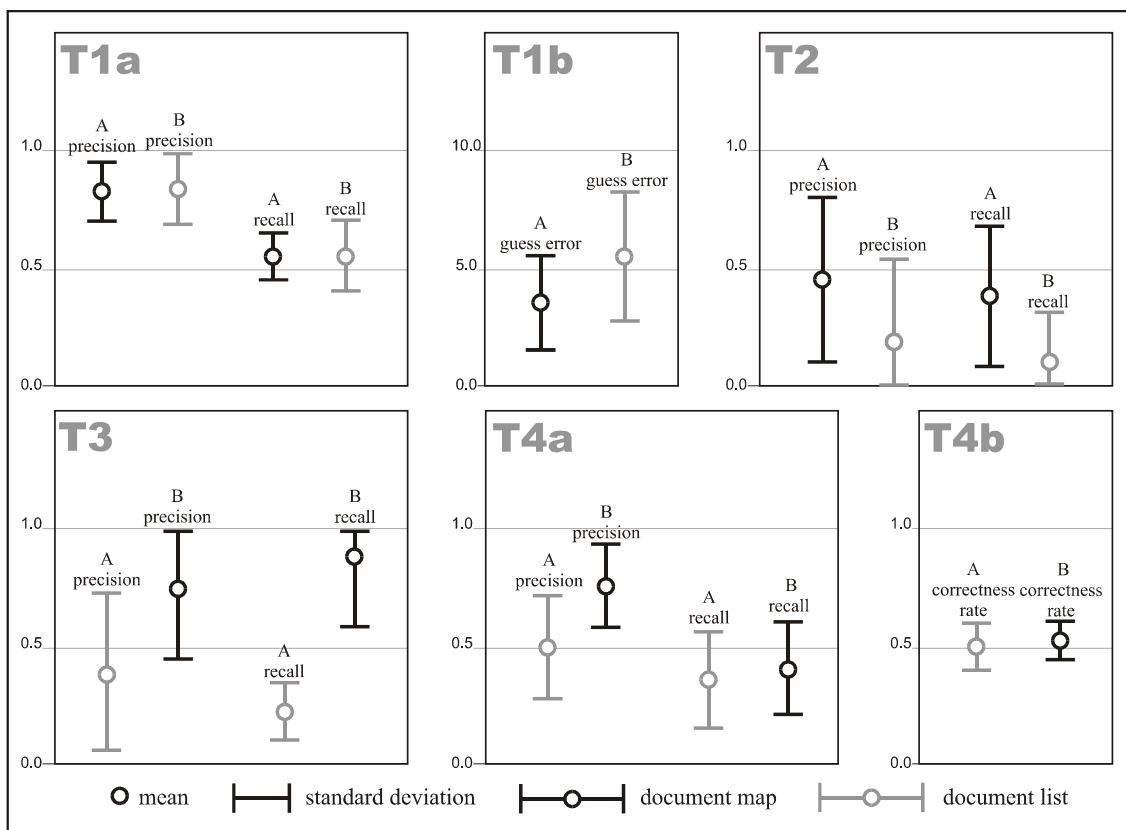


Figure 9-7: Box plot of average values of dependent variables (solution quality) for each task

Though language skills do not differ significantly between the groups we compared the results of non-native speakers with the average results of native speakers in group B for each task. However, no significant differences were found. Consequently, for interpreting the results presented in the following, group-specific considerations are not necessary.

9.4.3 Results of the Tasks

Table 9-10 presents the experimental results regarding the solution of tasks, i.e. empirical mean and standard deviation for the dependent variables which indicate quality of task solutions. As in all remaining tables the better mean is printed in bold-face. Figure 9-7 gives a box

Table 9-11: Results of statistical significance tests: Error probabilities α connected to test values (t , T , F , and U).

task	measure	Student's t -test (2-sided)	Welch's T -test (2-sided)	F -test (variance)	Wilcoxon-test (2-sided)
1a	precision	0.836	0.838	0.370	0.6286
1a	recall	0.681	0.686	0.161	0.9474
1b	guess error	0.081	0.083	0.658	0.0447
2	precision	0.066	0.068	0.705	0.0507
2	recall	0.019	0.018	0.319	0.0172
3	precision	0.005	0.005	0.587	0.0065
3	recall	< 0.0001	< 0.0001	0.001	< 0.0001
4a	precision	0.001	0.001	0.540	0.0019
4a	recall	0.531	0.532	0.746	0.7930
4b	correctness rate	0.370	0.368	0.677	0.2286

plot of the result data and shows that the respective group working with the document map system achieved better results for all tasks apart from the precision in task T1a. Furthermore, the quality differences are much clearer in tasks T1b, T2, T3, and T4a than in T1a and T4b.

Table 9-11 shows the results of the statistical trend tests. As in all remaining tables containing test results, probabilities less than 0.1 are printed in bold-face, probabilities less than 0.05 are shaded in addition (for the F -test all probabilities $\alpha < 0.2$ are highlighted). Since Student's t -test and Welch-test yield nearly identical results the variance can be neglected (even for T3 recall where the F -test indicates an inhomogeneity of variances). The Wilcoxon-test confirms the results of the other tests. Using the t -test results (Student), for task T1a, T4a (recall) and T4b we cannot reject the null hypotheses. However, we can reject the null hypotheses for the remaining tasks at least at 0.1-level, in detail:

- ($H_{0T1b,g}$) There is a tendency towards significant differences ($\alpha = 0.081$) between groups regarding guess error g for assessing the approximate number of documents for topics (T1b). Together with the difference in means (table 9-10) we can accept ($H_{1T1b,g}$) at 0.1-level.
- ($H_{0T2,p}$) There is a strong tendency towards significant differences ($\alpha = 0.066$) between groups regarding precision p for finding topical outliers in a collection (T2). Together with the difference in means (table 9-10) we can accept ($H_{1T2,p}$) at 0.1-level.
- ($H_{0T2,r}$) There are significant differences ($\alpha = 0.019$) between groups regarding recall r for finding topical outliers in a collection (T2). Together with the difference in means (table 9-10) we can accept ($H_{1T2,r}$).
- ($H_{0T3,p}$) There are significant differences ($\alpha = 0.005$) between groups regarding precision p for identifying classification errors in a collection (T3). Together with the difference in means (table 9-10) we can accept ($H_{1T3,p}$).
- ($H_{0T3,r}$) There are significant differences ($\alpha < 0.0001$) between groups regarding recall r for identifying classification errors in a collection (T3). Together with the difference in means (table 9-10) we can accept ($H_{1T3,r}$).
- ($H_{0T4a,p}$) There are significant differences ($\alpha = 0.001$) between groups regarding average precision p^* for associating related documents in a collection (T4a). Together with the difference in means (table 9-10) we can accept ($H_{1T4a,p}$).

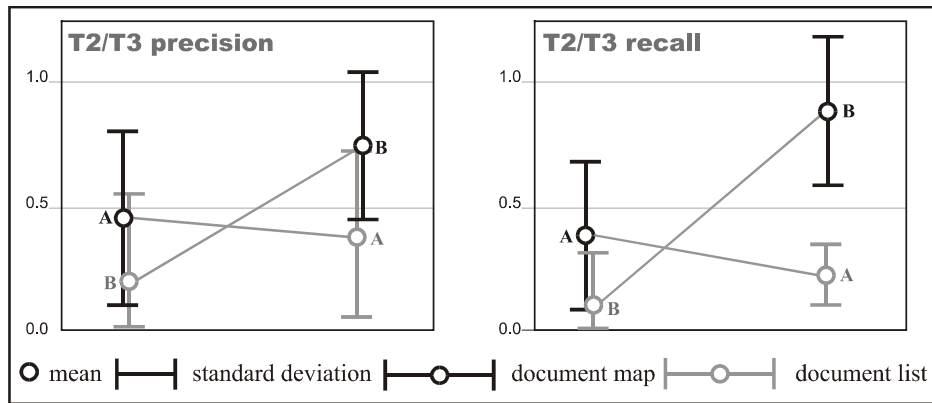


Figure 9-8: Crossover effect observed for tasks T2 and T3 after treatment change

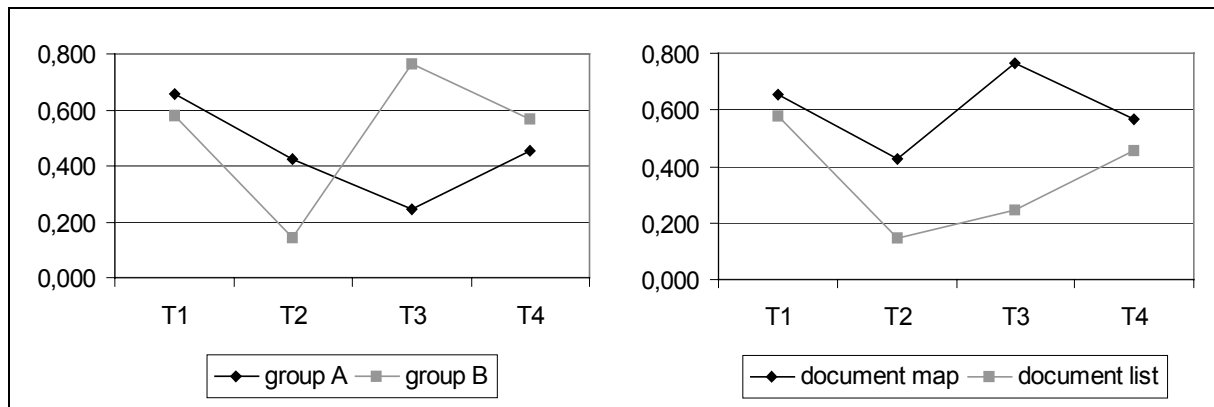


Figure 9-9: Overall performance of groups (left) and systems (right). The diagrams show the averaged values of task solution quality.

Considering tasks T2 and T3 we can observe a clear reversion of group performance after the system change (figure 9-8 depicts this situation). These two tasks are relatively similar: Both are quality control tasks in which the analyst has to use detailed information from documents, and in both cases there is a strong need to recognize irregularities within the collection. Once such irregularities are localized it is relatively easy to validate them. The crossover effect underpins the effect caused by the systems.

Figure 9-9 depicts the overall performance of each group or system, respectively. For this, the means of the results of the single quality measures for each task are averaged (without weighting a particular measure). Therefore, the average guess error g is inverted and standardized to the interval $[0,1]$ by $g' =_{\text{def}} 1 - g/g_{\text{max}}$. Both, the crossover effect and the superiority (at least in trend) of the document map system regarding the quality of task solutions can be seen.

9.4.4 Results of the Qualitative Questionnaires

9.4.4.1 System Transparency

After having performed each task the test subjects were asked to rate the overall quality of their results on a 5-item Likert scale (5 = very good, 1 = very poor). Based on this assessment is the transparency error $Q-U \in \{-4, \dots, 0, \dots, 4\}$, as defined in section 9.2.4.3, where 0 means optimal transparency and negative (positive) values correspond to overrating (underrating). Note that the transparency error is trivially optimal if a test subject did not find any solution and thus judges his or her solution quality to be very poor. Table 9-12 presents the results of the transparency error measure for all tasks and each single quality measure. Though test sub-

Table 9-12: Transparency errors for each task and each measure, results of statistical significance tests

		task 1			task 2		task 3		task 4		
		precision	recall	guess error ⁶	precision	recall	precision	recall	precision	recall	correctness
group A	μ	1,800	0,467	0,621	0,200	-0,133	0,333	-0,133	0,533	-0,133	0,533
	σ	1,082	0,915	1,374	1,373	0,834	1,759	0,990	1,407	1,457	1,246
group B	μ	2,429	1,000	0,800	0,429	0,071	0,107	0,679	1,286	-0,357	0,143
	σ	0,756	1,177	1,424	1,222	0,730	0,836	0,912	0,994	1,277	1,027
χ^2 -test		0,120	0,577	0,476	0,097	0,126	0,149	0,216	0,207	0,862	0,743
χ^2 -test on Q-U		0,032	0,176	0,837	0,014	0,015	0,129	0,699	0,966	0,851	0,928

jects rated their *overall* quality it is interesting to examine how this judgment corresponds to the single aspects of solution quality.

Regarding the corresponding hypotheses we have the following results: Only the null hypotheses for task T1 (precision), task T2 (precision and recall) can be rejected. In detail:

- (H0_{T1,p,transp.}) There are significant differences ($\alpha = 0.032$) between groups regarding the transparency error $Q-U$ for self-assessing the quality of achieved results with respect to precision p . Together with the difference in means we can accept (H1_{T1,p,transp.}).
- (H0_{T2,p,transp.}) There are significant differences ($\alpha = 0.014$) between groups regarding the transparency error $Q-U$ for self-assessing the quality of achieved results with respect to precision p . Together with the difference in means we can accept (H1_{T2,p,transp.}).
- (H0_{T2,r,transp.}) There are significant differences ($\alpha = 0.015$) between groups regarding the transparency error $Q-U$ for self-assessing the quality of achieved results with respect to recall r . However, together with the difference in means we have to reject (H1_{T2,r,transp.}) and to accept a corresponding alternative hypothesis in favor of the document list system.

9.4.4.2 Qualitative Judgments Accompanying Tasks

In addition to the self-assessment of solution quality the test subjects were asked, after having completed a task, to judge the system suitability for the assigned task and the effort for solving it with the system at hand on 5-item Likert scales. In addition, after task 1 we asked the test persons to judge their familiarity with the document corpus. Table 9-13 presents the results of these questions and evaluates trends in the data (though the answer distributions are rather mound shaped in nearly all cases, we applied a χ^2 -test in addition). Figure 9-10 visualizes the results and shows the trends.

For all tasks and questions (except system suitability for task T1) the user judgment favors the document map system in average: More precisely, members of group A clearly judge their familiarity with the text corpus higher than members of group B after task T1. There are no significant differences in the assessment of system suitability for task 1 and the effort for solving this task. The judgments regarding system suitability for tasks T2, T3, and T4 are significantly higher by test subjects working with the document map system, and the effort for solving these tasks is significantly judged to be lower by document map users (higher values in table 9-13 due to rating scale encoding). Regarding the formal hypotheses we have the following results:

⁶ Transformed according to $g' =_{def} 1 - g/g_{max}$.

($H_{0_{T1,fam.}}$) There is a tendency towards significant differences ($\alpha = 0.074$) between groups regarding the self-assessment of familiarity with the text corpus after having performed task T1. Together with the difference in means (table 9-13) we can accept ($H_{1_{T1,fam.}}$) at 0.1-level.

For tasks T2, T3, and T4:

($H_{0_{Ti,suit.}}$) There are significant differences ($\alpha_{T2} = 0.021$, α_{T3} , $\alpha_{T4} < 0.001$) between groups regarding the assessment of system suitability for the assigned task. Together with the differences in means (table 9-13) we can accept ($H_{1_{Ti,suit.}}$).

($H_{0_{Ti,effort}}$) There are significant differences ($\alpha_{T2} = 0.009$, $\alpha_{T3} < 0.001$, $\alpha_{T4} = 0.019$) between groups regarding the assessment of effort necessary for solving the assigned task. Together with the differences in means (table 9-13) we can accept ($H_{1_{Ti,effort}}$).

The question regarding self-assessment of quality was primarily used for the transparency measure. However, the pure rating itself provides interesting information: In average, test persons using the document map system rate their solution quality higher (but not necessarily more precisely, cf. section 9.4.4.1) than people using the document list system (for tasks T2 and T3 this difference is even significant).

9.4.4.3 Final Overall Preference Judgment

Following the solution phase of all tasks we asked the test subjects to name the system they

Table 9-13: Results of qualitative questions after each task

familiarity with corpus ¹⁾			system suitability ¹⁾				self-assessment quality ¹⁾				effort ²⁾			
task		1	1	2	3	4	1	2	3	4	1	2	3	4
group A	μ	3.067	3.267	3.267	1.714	2.786	2.733	2.667	1.929	2.571	3.267	3.333	1.929	2.429
	σ	0.594	0.799	0.884	0.726	0.975	0.961	1.175	1.072	0.938	0.799	0.816	0.917	0.938
group B	μ	2.643	3.286	2.393	4.286	4.071	2.143	1.429	3.607	2.929	3.143	2.214	3.857	3.286
	σ	0.633	0.914	1.041	1.139	0.616	0.663	0.646	1.077	0.917	0.864	1.251	1.167	0.726
t-test (Student)		0.074	0.953	0.021	<0.001	<0.001	0.113	0.002	<0.001	0.318	0.822	0.009	<0.001	0.019
T-test (Welch)		0.075	0.953	0.023	<0.001	<0.001	0.114	0.002	<0.001	0.318	0.822	0.009	<0.001	0.021
F-Test (variance)		0.810	0.623	0.550	0.117	0.110	0.237	0.033	0.985	0.937	0.790	0.166	0.492	0.318
χ ² -test		0.045	0.884	<0.001	<0.001	<0.001	0.274	0.001	<0.001	0.168	0.287	0.002	<0.001	0.016

rating scale values:
1) 5 = very good, 4 = good, 3 = moderate, 2 = poor, 1 = very poor
2) 5 = very easy, 4 = easy, 3 = moderately strenuous, 2 = strenuous, 1 = very strenuous

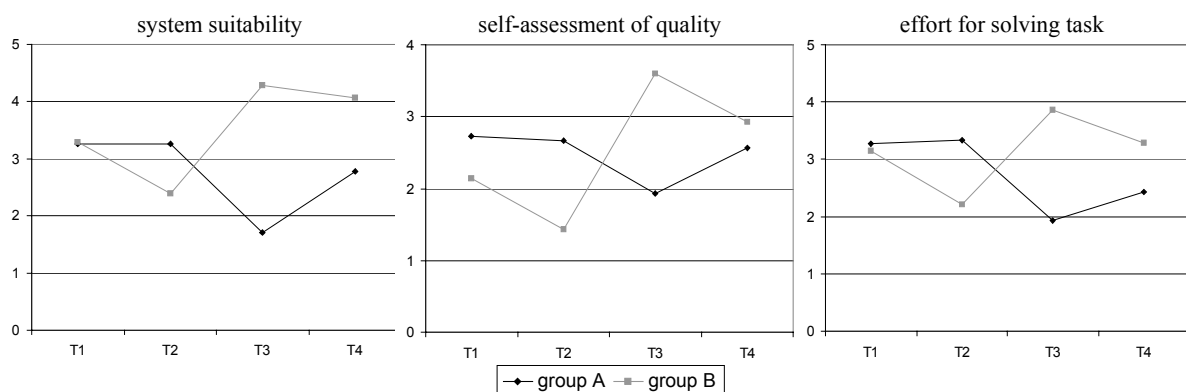
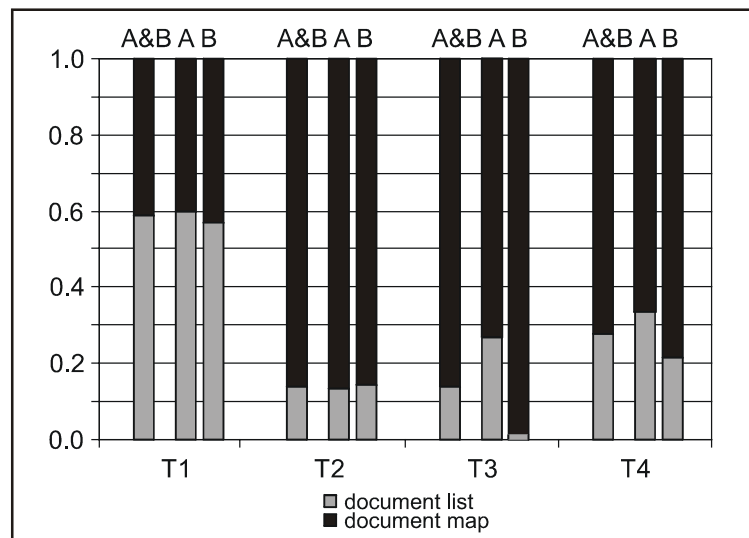


Figure 9-10: Visualization of means for qualitative questions after each task. Regarding the effort, note that a high value corresponds to a low degree of effort due to the uniform Likert scale encoding applied.

Table 9-14: Frequency distribution of task-specific system preference (final judgment, relative frequencies)

Which system would you prefer for each task?		T1	T2	T3	T4
groups combined	document map	0.414	0.862	0.862	0.724
	document list	0.586	0.138	0.138	0.276
group A	document map	0.400	0.867	0.733	0.667
	document list	0.600	0.133	0.267	0.333
group B	document map	0.429	0.857	1.000	0.786
	document list	0.571	0.143	0.000	0.214
comparison of groups (χ^2 -test)		0.739	0.782	< 0.0001	0.298

**Figure 9-11: Relative frequencies of answers to preference question for each task (groups combined and comparison of single groups)**

would prefer for solving each single task. Since both groups worked with both systems and due to the relatedness in nature of tasks T1 and T4 as well as T2 and T3, test subjects could at least perform an indirect comparison of the tools. The results of this preference question is given in table 9-14, figure 9-11 visualizes the relative frequencies of answers. For tasks T1, T2, and T4 both groups are in agreement regarding the judgment (there are no significant differences between groups): Altogether, 58.6% of test subjects would prefer the document list system for task T1, only 41.4% would use the document map. For tasks T2 and T4, a clear preference for the document map system can be observed (86.2% and 72.4%, respectively). Regarding the preference for task T3 both groups have different judgments (significant group difference with $\alpha < 0.0001$): Only 73.3% of members of group A (who worked with the document list system for T3) would rather use the document map instead, but all members of group B would use the document map system again for this task.

In addition, we asked the test subjects how each system appealed to them. For both questions there are no significant differences between the judgments of both groups (according to the χ^2 -test we have $\alpha = 0.628$ for the question about the document map and $\alpha = 0.663$ for the question about the document list). Figure 9-12 visualizes the respective answer distribution. Both distributions differ significantly ($\alpha < 0.0001$ according to the χ^2 -test). For rating scale values 5 = very good, 4 = good, 3 = moderate, 2 = poor, 1 = very poor, the mean for the document map distribution is $\mu = 3.966$ ($\sigma = 0.626$), the mean for the document list is $\mu = 2.655$ ($\sigma = 0.857$), i.e. the document map is clearly preferred so that ($H_{0_{\text{pref.}}}$) must be rejected and ($H_{1_{\text{pref.}}}$) can be accepted.

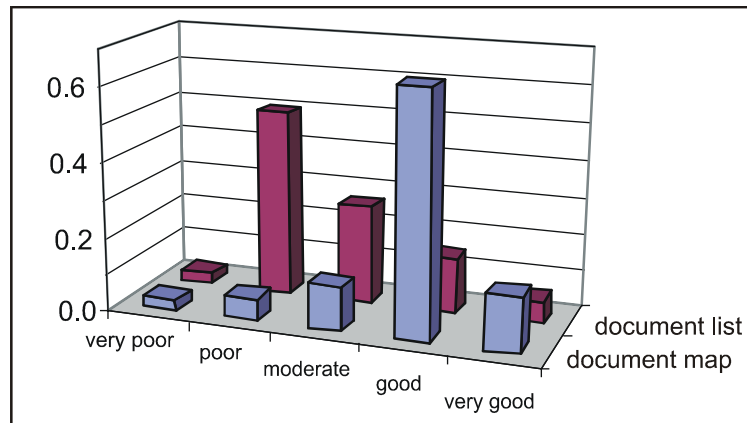


Figure 9-12: Visualization of answer distribution for the question “How did working with the document map system (document list system) appeal to you?”

9.4.5 Observed Correlations Between Variables

In this section the correlation coefficients between all value series for variables are considered. Table 9-15 presents the correlation coefficients for quality values of the task solutions. Coefficients above 0.5 (absolute value) are printed in bold-face, values above 0.75 are shaded in addition. Besides from the weak negative correlation between ‘T2 recall’ and ‘T3 recall’ (note the system change between these tasks) and the weak positive correlation between T3 recall and T4a precision there are no clear correlations among different tasks, thus underpinning the theoretical discussion on the independence of tasks (cf. section 9.2.6.6). Only within tasks T2 and T3 there is a clear positive correlation between precision and recall (cf. discussion in section 9.5). Table 9-16 shows the correlation coefficients above 0.75 between values of remaining variables.

Table 9-15: Correlation coefficients for quality values of task solutions regarding the single measures

task	measure	1a	1b	2		3		4a		4b
		recall	guess error	precision	recall	precision	recall	precision	recall	correctness rate
1a	precision	-0.17	0.05	0.02	0.11	0.04	-0.15	0.17	0.18	0.10
	recall		0.23	0.13	0.18	-0.08	0.00	-0.03	0.23	0.02
1b	guess error			-0.08	-0.16	-0.10	0.10	-0.07	-0.03	-0.08
2	precision				0.85	-0.10	-0.46	-0.07	0.28	0.00
	recall					-0.24	-0.53	0.02	0.14	-0.05
3	precision						0.77	0.38	0.29	0.14
	recall							0.52	0.25	0.12
4a	precision								0.07	0.15
	recall									0.18

Table 9-16: Correlation coefficients above 0.75 between values of remaining variables

variables		correlation coefficient
T3 system suitability	T3 recall	0.92
T3 system suitability	T3 effort	0.90
T3 system suitability	T3 self-assessment quality	0.85
T3 transparency	T3 recall	0.82
T3 effort	T3 recall	0.81
T3 effort	T3 self-assessment quality	0.81
T2 recall	T2 self-assessment quality	0.79

9.4.6 Comments on Systems by the Test Subjects

After the experiment the test persons were asked to name the greatest advantage and the greatest disadvantage of each system (free comments). Table 9-17 summarizes given answers which have been named at least 4 times. Note that this compilation is not exact as we had to cluster and interpret free statements (not multiple choice questions). However, the list gives a qualitative impression of the reception of the different systems.

Table 9-17: Classification (rough) and approximate frequency of free comments regarding pros and cons of both systems

document map	pros	Spatial arrangement of texts (similarity information) is helpful.	••••••••••
		Global overview of text collection can be obtained easily.	••••••••
		Texts can be easily grouped regarding their contents.	••••••••
		The graphical presentation is appealing.	••••
	cons	Overview of text titles is difficult.	••••••••
		Using a map display is strange.	••••••••
		Assigning document points to texts or titles is difficult.	••••••
		Getting an overview of topics is difficult.	••••
document list	pros	Titles are clearly arranged and easily/quickly available.	••••••••••
		Titles are a good indication for the documents' content.	••••••••
	cons	Presentation of list is badly arranged for many texts.	••••••••••••••
		Texts are not clustered according to their contents.	••••••••
		Relatedness of texts is difficult to recognize.	••••••••

9.4.7 Results of the Interaction Protocols

The experiment control system in which both document access systems were integrated recorded every interaction step carried out by the users along with time-stamps and relevant parameters. Figure 9-13 shows the number of actions of different types: The total averaged numbers of actions (including selecting or opening documents, querying, change of dialogues and presentations, but also other mouse and keyboard actions) are given in the bar charts. The action classes depicted in the pie charts comprise only the most important interactions.

Due to the layout of the system interfaces there is always a certain dialogue register open. Figure 9-14 presents pie charts for every group and task which show the proportion of time for which the respective registers were open. Though both figures are related to each other there is one important difference: The second figure may also include passive or recognition phases whereas the first only shows actual interactions. But since the registers may show important information (query results, keywords, important sentences,...) the time spent in different dialogues is of interest for an interpretation of the way the different systems are used for the single tasks.

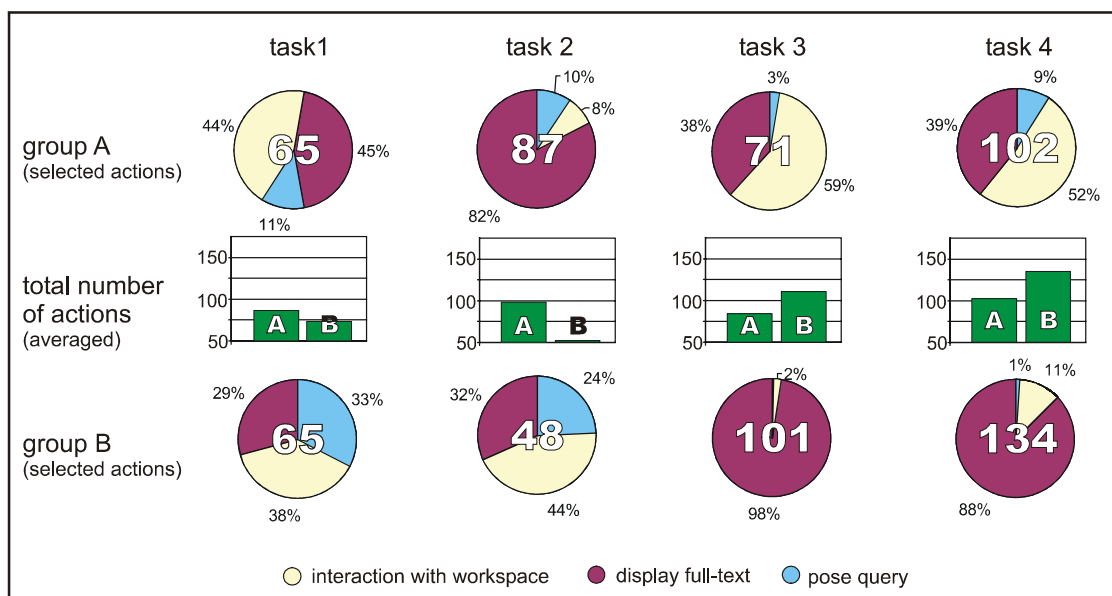


Figure 9-13: Average number of interaction carried out by the test persons for both groups and each task, relative frequencies of actions of selected types (total averaged number of considered actions is given for each pie chart)

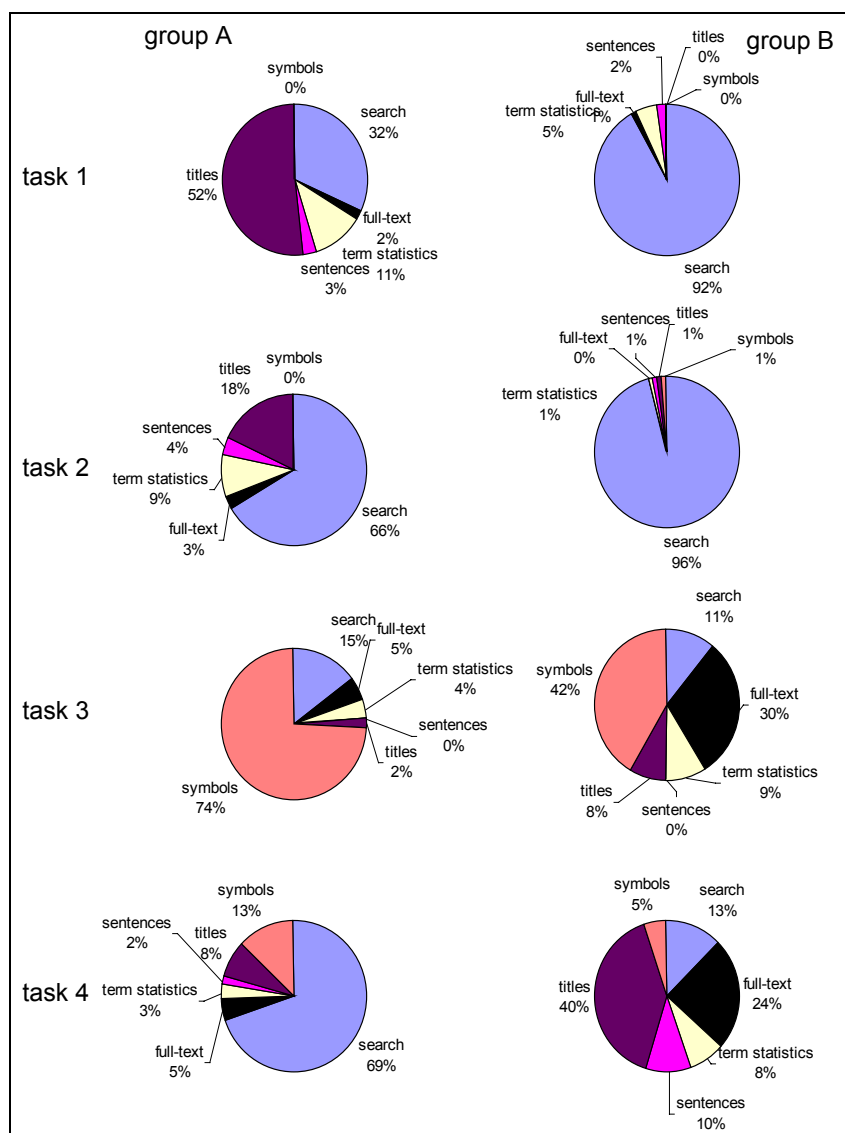


Figure 9-14: Average proportion of time for which different registers were visible

9.5 Interpretation and Conclusion

The experiment clearly confirmed most of the hypotheses about the adequacy of document maps for corpus analysis tasks in knowledge management: The computation of the overall similarity structure of the text corpus and its visualization helps to significantly improve the effectiveness of task solutions. Furthermore, test subjects would prefer the document map system in nearly every case. The following sections sum up the major results and discuss their implications for the understanding of the support offered by document maps for managing the knowledge contained in specialized document collections.

9.5.1 Interpreting the Results for Each Task

9.5.1.1 Getting an Overview of a Collection's Subject Structure

Regarding the first part of this task, T1a (identifying topics), the means and standard deviations for precision and recall of both groups are nearly identical. Test persons confirmed that the titles of the documents gave pretty good hints about their contents so that just scanning them was a good strategy for this task. Some persons found it hard to get an overview of the topics using the document map because the text titles were only loosely coupled with the graphical presentation. The additional similarity information provided by the map display was no advantage (regarding the proposed quality measures). Part of the reason for this may be, however, that the text titles were very descriptive, which is a genre-dependent feature and cannot be generalized. In other contexts the text clustering might be more advantageous which, of course, still has to be shown.

The second part of the task (T1b) was concerned with guessing the 'strength' of each topic in the collection. For this task we could observe a very clear tendency that document map users work out better solutions. In contrast to T1a, this part of the job required a deeper understanding of the text collection's overall structure – an aspect which obviously is supported well by the graphical similarity display.

In summary, though there is no difference in quality for T1a, in total the document map group performed more effectively for this task: Though the map was not of additional help for identifying topics it seems to be of use for effectively grasping structural aspects of the collection. Moreover, other advantages of the document map could be shown: In tendency, the transparency of working with the document map was better, for the precision in T1a even significantly. Furthermore, there is a clear tendency that document map users feel more familiar with the corpus after getting an overview with this tool, which is an indication for the clearness of the map display.

Both groups judge their tool to be of medium suitability for solving the task (nearly identical means and standard deviations). Looking at the interaction protocols shows – as we expected – that the document list users make clearly more use of the query interface: The search dialogue was opened 3-times as long as for the document map, and 3-times more queries were posed. Document map users displayed the register with text titles more than half of the interaction time. In other words, the document map seems to support a rather passive recognition of topics and their strengths whereas the list system requires more active work by specifying queries.

Despite the objective superiority of the document map system (at least in clear tendencies), members of both test groups were in agreement with their final preference judgment: More than half of the test persons (nearly 60%) would prefer the document list system for solving

this task. The reason for this might be derived from the free comments of the test persons on the systems (cf. section 9.4.6): The map was not only new to the users and required some habituation, but a stronger connection of graphical display and textual content would be appreciated. Though the full research prototype has some simple additional functions for a text-graphic connection (inscribing document symbols with titles, field description) which were excluded from the experimental system, a suitable integration of descriptive text parts and graphical similarity information is still missing. A closer integration of the ‘best of both worlds’ is an interesting research question, the relevance of which was emphasized by the results of this task.

9.5.1.2 Checking the Clarity of the Semantic Structure of the Collection

Task T2 dealt with finding outlier documents within a collection. For this task it turned out that the graphical display of the similarity structure clearly helps to quickly find such topically isolated texts. We observed a very clear tendency regarding precision of achieved solutions and a highly significant result regarding recall in favor of the document map. Whereas the transparency of the document map is significantly better regarding precision, for recall the document list system achieved significantly better results. This, however, is due to the fact that solutions were hardly found by the document list group and thus test subjects could trivially assess their solution quality to be very poor (which corresponds to the objective results observed). Obviously, the transparency measure is more interesting for cases in which possible solutions of the tasks were produced at all.

As section 9.4.5 has shown, precision and recall of solutions for this task are clearly correlated (to a degree of 0.85). An interpretation for this is that possible outlier candidates, once they were localized, could be verified rather easily, not least because the collection was compiled in a way that outlier texts undoubtedly were off topic. The judgments regarding system suitability are significantly better for the document map system than for the text list, and the document map group stated a significantly lower effort for solving the task.

The document map group was plainly more ‘active’ during this task (100 vs. 55 interaction steps in average). A more detailed view on the types of actions performed reveals that map users opened clearly more text documents than the list group who worked rather query-oriented. In most of the time the query interface was opened by the document list users. In relation to the total number of actions the text list users more frequently interacted with the workspace. The map group spent more time for looking at other helpful information, such as significant terms and important sentences. Taking these observations and the good solution quality, apparently the document map system allows a goal-directed reading (cf. section 8.2.4) and verification whereas the comparative system requires a time- and effort-consuming localization phase. Accordingly, there is a clear preference of test subjects to the document map (86%), independent from group membership.

9.5.1.3 Checking the Correctness of a Classification of Text Documents

In this task (T3) we could observe that both, precision and recall are clearly better for the document map users (with a very high statistical significance). As in T2 we have a clear correlation between precision and recall. The interpretation is similar to that from above: Once a possible classification error was localized it was very easy to verify if the found text indeed was misclassified. Regarding transparency there are no significant differences between groups. Highly significant differences, however, can be observed in the assessment of system suitability: The document map users judge their system to be well (with a tendency to ‘very well’) suitable whereas the list users assess their tool to be poorly (with a tendency to ‘very

poorly) suitable for solving the task. Document map users found solving the task easy. In contrast, list users had to try hard to solve it (with a highly significant statistical difference).

Looking at the interaction protocols it is noticeable that document map users opened clearly more full texts than the list group, which suggests that the map leaves more time for actually coping with the documents' contents due to a relatively short localization phase. In their final judgment all test subjects preferred the document map. In particular, all persons who used the map for this task would use it again, though not all document list users are convinced that the map is really superior.

9.5.1.4 Associating Related Documents and Understanding their Semantic Context

The first part of this task, T4a, concerned with grouping related documents, could be solved with a significantly higher precision by the document map group than by the document list users. Regarding recall, no significant differences between groups could be found. The same holds for the correctness rate of knowledge questions in T4b. The latter may be caused by a design flaw, since the questions asked turned out to be very complicated and too fine-granular so that most test subjects were forced to guess an answer instead of using acquired knowledge (which many of them remarked after the study).

The reason that we still believe that map users should be more familiar with the texts' contents – though we failed to show it empirically – can be found by considering the interaction protocols: The document map group (group B) displayed clearly more full texts (88% of the considered actions) than the comparative group (only 39%). The time spent in different register dialogues shows that map users more often displayed content-relevant information (term statistics and important sentences) than list users who primarily searched the collection using the query interface (9.2 queries posed and search register open in 69% of the time in contrast to 1.3 queries and 13% time for the map users in average). Together with the precision result this indicates that the document map allows to find related documents more quickly and more precisely and spares the users enough time to deal with the contents of relevant text groups instead of searching around.

The document map users rated their tool significantly higher than the document list group, and map users found solving the task significantly easier than list users. In total, test subjects clearly prefer the document map for this task (more than 70% of all test persons).

9.5.2 Overall Conclusion

For all tasks performed in the experiment the document map turned out to have at least some clear advantages. Not a single results is clearly against the map (apart from the transparency artefact for T2). Taking together all quality results, the document map group performed better than the document list group for all tasks. In summary, working with the document map was significantly more appealing to the test persons than working with the list system. The document map clearly makes easier the access to complex specialized document collections and enhances the effectiveness of solving real-world knowledge management tasks. The clear crossover effect for the system change between tasks T2 and T3 indicates – together with the very clear quality differences of groups – that the map is especially suitable for identifying structural irregularities, and test persons confirm that the graphical spatial arrangement is helpful for getting a global overview of the collection.

However, we also found important drawbacks of the map display which are mainly concerned with the loose integration of textual information into the workspace: People would like to see more text, tightly coupled to the graphical display or another expressive arrangement of text

clusters (cf. [Hea99], p. 274 ff., see also section 11.3.3). How such a combination should ideally look like and how it can be realized technically seems not to be answered sufficiently yet. In any case, taking into consideration the quality of results achieved by test persons with the graphical map display in this study, it seems to be fruitful to strive for a close coupling of text and graphical arrangement. This is especially true since the map seems to leave more time for coping with contents, so we should find ways of integrating more semantic contents of the text into graphical displays without losing the benefits of both.

One general lesson learned in this study is that empirical investigations of the performance of interactive information systems – especially those concerned with the complex field of text retrieval – provide a rich source of information for computer science researchers and raise important research questions which can significantly improve the understanding of how novel systems must be designed in order to support complex real-world tasks. This assessment can also often be found in literature. Further task-related studies of using the proposed document map approach for corpus analysis in knowledge management – and of visual text-retrieval and text-access principles and tools in general – will surely point to additional and promising research possibilities (see also section 11.3.1).

10 **Extension of the Basic Framework: Incorporating Adaptability**

A document map approach based on the basic framework as introduced in chapter 6 performs an automatic structuring of a collection of documents. The similarity or dissimilarity between documents is computed by a suitable document analysis module which can be chosen with respect to requirements of the application domain. In this sense the structuring performed by a certain realization of the framework can be seen as an ‘objective’ structuring which has to be interpreted by the human analyst. This chapter deals with the question of how to integrate a ‘bias’ towards a special interest, so that analysts can influence a document map regarding their personal interest. Technical details have been implemented in [Tusk00].

10.1 **A Rule-Based Approach for Incorporating Adaptability**

In chapter 5.8.4 the need for an adaptable document map approach has been motivated. According to this motivation, in this work the idea of ‘adaptability’ is to allow the analyst to influence the map generation process by incorporating a ‘bias’ towards his interest and by providing scenario-related background knowledge. A ‘bias’ in this sense considers both, a general understanding of document relatedness and a personal interest: In addition to a priori given document relationships it determines a degree of relatedness between documents and smoothly incorporates this information into an existing document map. Such a setting is useful when the analysis is less focused in content and excluding potentially helpful semantic connections would impede the analysis process.

Figure 10-1 depicts the considered situation: The document map is seen as being influenced by both, the automatic pre-structuring (which uses domain-adequate document analysis methods) as well as the personal interest and the background knowledge of a domain expert and analyst. The pre-structuring process extracts information about the relatedness of documents from the collection in an automatic way by using a pre-defined document similarity measure. The analyst has domain-specific background knowledge and a certain impression about the documents’ contents and wants to form a bias towards his interest. But how can a bias be incorporated into the map so that the personal interest is reflected in the final structure to be visualized? From a user’s point of view the desired bias can be realized by the definition of *rules* that contribute background knowledge and weighting of certain task- and domain-dependent key-concepts. These rules define additional aspects of document relatedness (and thus model the desired bias) and have to be considered by the automatic map generation process. Then the collection can be structured accordingly, shifting focus towards the analyst’s interest by stressing certain similarities or weakening others. As a result a user’s specific interest field is incorporated into a document map, allowing the definition of different perspectives on the collection.

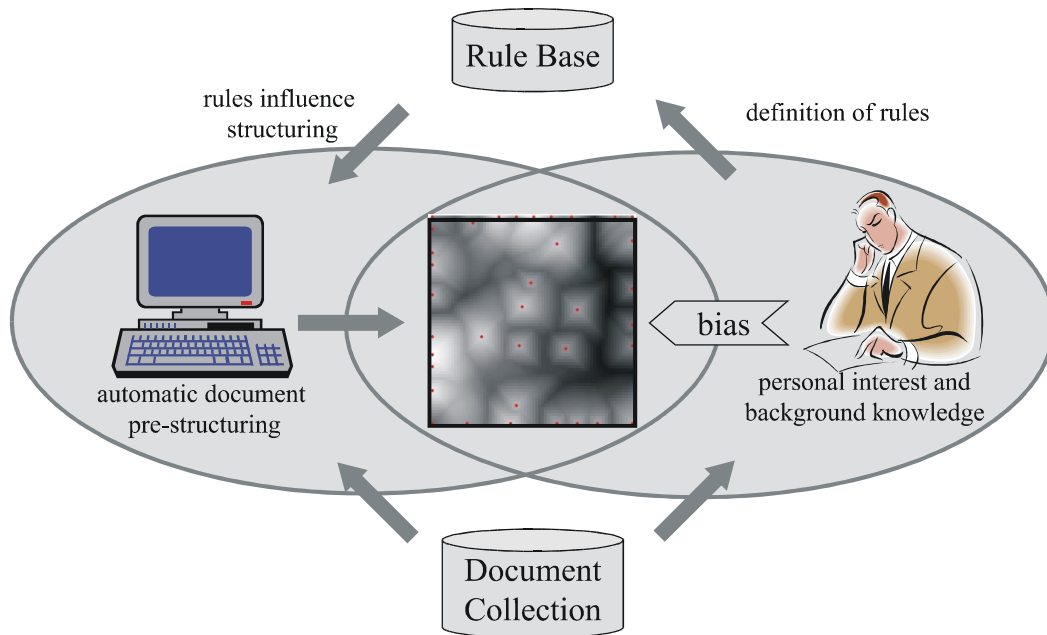


Figure 10-1: Document map in the sphere of influence of automatic pre-structuring and the analyst's personal interest

As a side-effect note that there is also an alternative way to take advantage of such a rule-based approach: Though there are scientific approaches for domain-specific document analysis and comparison methods (cf. chapter 3) their application in practical environments is still a matter of availability and efficiency. The adaptation of domain-specific methods is expensive and not always possible. From a practical point of view it is thus desirable to apply simple and available domain-independent and resource-saving methods first. If the pre-analysis as performed by the given document analysis module is not sufficient in all necessary aspects, a rule-based approach could be used to refine the achieved pre-structuring by making use of some additional background knowledge. Note that for practical reasons an extensive knowledge base should not be required, i.e. it is often not practicable to compare documents only by using a knowledge base because that would mean an enormous effort of knowledge base implementation and maintenance. In this sense it is useful to strive for a hybrid approach in which a pre-structuring method which does not require manual effort is the basis for a knowledge-based refinement of structures.

10.2 General Design Decisions

There are two basic questions which have to be addressed on the way towards the desired rule-based approach. The first question regards the extended architecture of the framework for generating document maps: Assumed that there is a satisfying way to model rules, how can an additional component for personal views (called 'semantic refinement module' in the following) be integrated into the basic approach? Section 10.2.1 motivates and sketches the answer to this question, the detailed solution is tackled in section 10.3. Having solved the problem of integrating a semantic refinement component into the framework, the second question is how to model rules that define document relationships in the context of the application domain 'managing the knowledge contained in specialized, especially technical document collections'? How would those rules be matched against the actual documents? Some general design decisions for this question are discussed in section 10.2.2. The detailed solution is more complex and is tackled in several steps: First, section 10.4 motivates the proposed design of

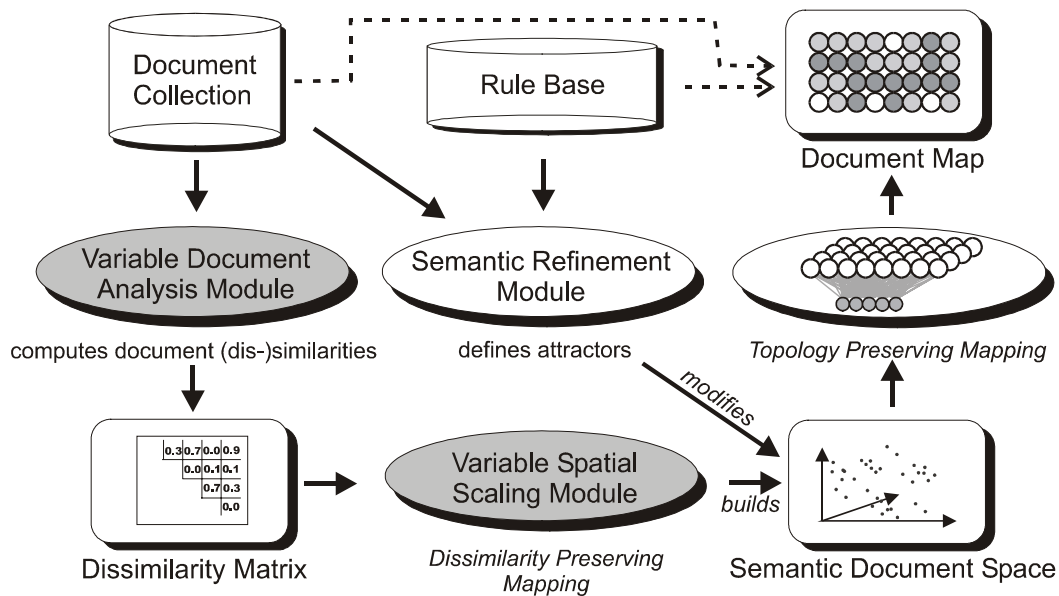


Figure 10-2: Integration of a semantic refinement module into the basic framework

rules. After that some basic techniques which are necessary for the rule approach are discussed (section 10.5) before section 10.6 presents its realization.

10.2.1 Integration into the Basic Framework

Given that a rule based approach defines a degree of relatedness of each pair of documents beyond the a priori known similarity, which interfaces of the basic framework allow an integration of this information? The main goals are to maintain the strict modularity of the basic framework and the exchangeability of the document analysis module, and to allow an intuitive understanding of the rule effects. Considering the general architecture of the basic framework (cf. chapter 6.1), there are two potential starting points for the desired integration: manipulating the dissimilarity value of each considered document pair in the dissimilarity matrix or directly influencing the object arrangement in the semantic document space. The first idea is obviously a problematic one: Among other problems, the manipulation of single document dissimilarities is likely to produce a significant local stress (cf. section 6.2.2). Due to the complex optimization problem of constructing the semantic document space based on dissimilarity information the result of this change of values is unclear. Thus, the structure would be changed only indirectly and the effects can hardly be predicted.

Considering the alternative, how can the arrangement of document representatives be influenced by a rule-based approach? The starting point for the following consideration is the semantic document space in which document representatives are placed according to the similarity of the corresponding documents. The placement of document representatives in this space, induced by the initial document analysis, contains important information on the document collection's structure. This structure shall be manipulated with the goal that the resulting placement reflects both, information derived by the pre-structuring and information which is 'superimposed' by the rule base. The idea pursued here is that rules define 'attractive forces' between document representatives, so-called attractors. Depending on the strength of the attractors the objects in the semantic document space will be moved, and thus arranged according to the defined bias. This approach brings along some interesting properties: First, it is a direct way to manipulate the given structure of the semantic document space. The effect can be understood more easily because it takes place in a familiar and transparent data structure.

Second, influencing the degree of relatedness of two documents implicitly affects the proximity to all other documents and establishes new relationships that depend on the original structure of the given document space. In other words, the given pre-structuring is taken into consideration. Third, document relationships as defined by ‘rules of relatedness’ do not necessarily have to be symmetric. Thus, it becomes possible to define rules which cause a movement of some documents or document groups towards others which remain more or less in place. This allows a more differentiated definition of a bias.

In order to maintain the framework’s modularity and the exchangeability of the document analysis component, the demanded semantic refinement module must not rely on features of a certain document analysis module. It may access only the document collection itself and has to have its own formal document representation. The resulting extended architecture of the document map framework is depicted in figure 10-2: The semantic refinement module directly operates on the given document collection. It is supplied with rules from an external rule base and calculates an additional degree of document relatedness. This information is used to modify the semantic document space which has been initialized (i.e. pre-structured) by the actual document analysis module. The semantic refinement module itself should comprise two clearly separated sub-components: a module for applying rules and calculating the additional document relatedness and a module for moving the objects in the semantic document space, so that this integration is independent of the rule component’s realization.

10.2.2 Design of the Rule Approach

Regarding the second question, how should the modeling language for rules look like? Since the users themselves will have to define rules rather than a knowledge engineer the proposed language has to be simple and intuitive enough so that the formal burden is not too high. Yet, the modeling elements should be expressive so that interesting relationships can be defined. A requirement for an application of an adjustable document map technique in a real-world knowledge management context is the demand for a staged approach: Constructing a large-scaled and complicated knowledge base is time-consuming and expensive – and beyond the idea of expressing a task-dependent bias. Thus, for pragmatic reasons there should be simple and flexible means of modeling rules as well as more elaborated and expressive ones which improve the rules’ precision, so that the user can decide how much effort he would like to invest. Note that – if rules are defined in a structured way – rule bases or parts of them can be re-used for similar analysis tasks or combined and re-assembled for different tasks.

Furthermore, of course, the rule language has to possess formal semantics so that the user unequivocally can understand the effect of the rule definitions. In section 10.4 the overall design of some useful rule types is motivated (without claiming completeness); the rules’ formal semantics are presented in chapter 10.6. The rules themselves should be designed close to the idea of moving document representatives in the semantic document space by attractive forces. The proposed selection of rule types is an application-oriented example which can be extended in a later work if the principle proposed here turns out to be useful. But that requires a modular design of the rule component: There should be a clearly specified ‘interface’ for rules so that new rule types can be defined and integrated easily, if desired. Finally, the methods applied for matching rules against documents should be efficient and sufficiently robust for simple document matching, so that real-world document collections can be processed in reasonable time.

10.3 Moving Document Representatives by Attractive Forces

In order to better understand the effect that the definition of rules will have on the semantic document space the modification of the space's structure will be discussed first. This section introduces the proposed method for moving document representatives according to attractors.

10.3.1 The Attractor Concept

As sketched in section 10.2.1, a rule-based approach shall result in some kind of measure which defines a degree to which documents are related with respect to the rules. This degree of relatedness shall be superimposed with the structure of the semantic document space. The idea for realizing such an approach is to move the document representatives according to 'attractive forces', called attractors, which result from applying rules to the collection of documents under examination.

Moving document representatives according to calculated relationships is a non-trivial problem. Document relationships cannot simply be interpreted as physical forces which cause an accelerated movement, since a corresponding physical system would never reach a balanced state and remain in movement or simply collapse. There are some methods which apply force laws from physics for information visualization. More precisely, these approaches interpret object relationships as physical forces and try to find a force-balanced state which reflects the given relationships: Spring embedding techniques model object relationships by mechanical springs of different strength and rest length. The aim is to obtain an object arrangement which best reflects the given relationships (cf. chapter 4.1.3). This approach is used for drawing weighted graphs, for example, where the springs' length depend on the weights of edges (cf. [Ead84, KaKa89, FrRe91]). The force-balanced state that such a spring system results in is generally independent of the initial state. However, dependency on the initial state is a basic requirement for the document map approach, since the original document relationships already encoded in the semantic document space shall be reflected in the final state. To avoid this it would be possible to model the initial state with the help of a spring system which is then modified according to the calculated attraction forces. Though, this would result in a system where the movement of a single object influences all other objects, even if there are no attractions defined for those objects. In another approach of force-directed placement (which uses a different force law) objects are moved under the influence of attractive and repulsive forces (cf. sections 5.4.1 and 5.4.3). Here, however, the resulting object arrangement is highly sensible to the starting conditions since slight changes in the initial state may cause clearly divergent final states which is again an undesired effect.

Therefore, the method proposed in this work is a vector translation approach which makes use of a simple physical metaphor (not a physical force law): An *attractor* can intuitively be seen as a positive force that specifies how strongly two document representatives (simply called objects in the following) in the semantic document space attract each other. In a physically lax phrasing, this force causes a movement of the mutually attracting objects towards each other. In this physically simplified metaphor attracted objects move towards each other in the semantic document space for a certain period of time. Their velocity depends on the degree to which the corresponding documents are related to each other as computed by the application of the given rules. The resulting arrangement of document representatives will reflect the relatedness of each pair of documents under consideration of the initial arrangement. Two basic demands for the desired object movement approach based on vector translation have to be taken into consideration:

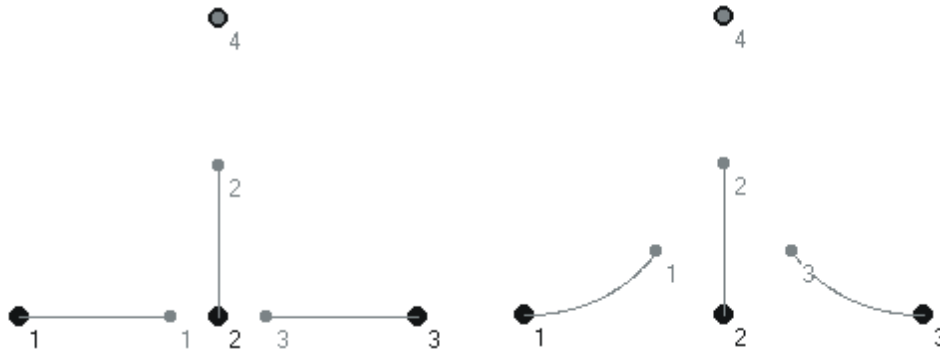


Figure 10-3: Object movement based on vector translation (a) without re-orientation and (b) with re-orientation. In either case object 2 attracts objects 1 and 3. Object 2 is attracted by 4. The initial arrangement is drawn in black, the resulting in gray.

1. Objects that attract other objects may be in motion themselves because they are attracted as well. Thus it is necessary that attracted objects follow those objects by whom they are attracted on their way. Figure 10-3 depicts this situation.
2. All object movements have to be executed simultaneously. Otherwise, the resulting configuration would depend on the order of vector translations.

10.3.2 Realization of the Proposed Attractor Approach

In order to proceed, a formal definition of ‘attractors’ and ‘attractor execution’ is necessary. An *attractor* $\phi_{i \leftrightarrow j}$ is modeled by a tuple of real numbers, i.e. $\phi_{i \leftrightarrow j} =_{\text{def}} (\phi_{ij}, \phi_{ji}) \in [0,1]^2$. An informal interpretation of $\phi_{i \leftrightarrow j}$ is that for a given pair of document representatives \mathbf{x}_i and \mathbf{x}_j the component ϕ_{ij} describes the strength with which \mathbf{x}_i is ‘attracted’ by \mathbf{x}_j and ϕ_{ji} describes the strength with which \mathbf{x}_j is attracted by \mathbf{x}_i . The totality of all attractors is defined by an *attractor-strength matrix* $\Phi = (\phi_{ij})$, $1 \leq i, j \leq n$, $\phi_{ij} \in [0,1]$, where $\phi_{ij} = 0$ if $i = j$. The attractor-strength matrix Φ stores information about the relatedness of documents regarding the rules that are defined in the rule base. The proposed movement of documents is realized by a vector translation method which is presented in the following.

According to the basic framework the semantic document space X is an m -dimensional metric space where each of the n documents of the given collection is represented by a real-valued location vector \mathbf{x} , i.e. $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \mathbb{R}^m$. If a document representative \mathbf{x}_i is attracted by \mathbf{x}_j , i.e. $\phi_{ij} > 0$, then \mathbf{x}_i moves towards \mathbf{x}_j , or more precisely, \mathbf{x}_i is shifted to some extent into the direction of \mathbf{x}_j . There might also be an attraction caused by other documents, i.e. for some $k \in \{1, \dots, n\}$, $k \neq i$, the attractor component $\phi_{ik} > 0$. Let $N(\mathbf{x}_i)$ denote the number of non-zero attractor components ϕ_{ij} for \mathbf{x}_i , i.e. $N(\mathbf{x}_i) =_{\text{def}} |\{j \mid 1 \leq j \leq n, \phi_{ij} > 0\}|$. Then, the resulting direction and ‘movement strength’ of \mathbf{x}_i should simultaneously depend on all attracting document representatives \mathbf{x}_j . The direction of movement has to be time-dependent: Due to the overall movement of objects the location of ‘attracting’ \mathbf{x}_j changes. In order to guarantee that document representative \mathbf{x}_i follows the movement of its attracting objects the direction of movement would have to be constantly adjusted. This adjustment is approximated by an iterative approach: For a total of t_{\max} iterations, in each iteration step t , $1 \leq t \leq t_{\max}$, only a certain fraction of the overall movement length is executed, so that the correct direction can be recalculated in the next step. Let $\mathbf{x}_i(1) =_{\text{def}} \mathbf{x}_i$ for $1 \leq i \leq n$. Then, the resulting movement strength and direction of \mathbf{x}_i at time t is defined by

$$\mathbf{v}_i(t) =_{\text{def}} \frac{1}{N(\mathbf{x}_i)} \cdot \sum_{j=1}^n \phi_{ij} \cdot \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{|\mathbf{x}_j(t) - \mathbf{x}_i(t)|}, \quad (10-1)$$

where $|\mathbf{x}|$ denotes the length of vector \mathbf{x} ($|\mathbf{o}| =_{\text{def}} 1$ for the zero vector \mathbf{o}). Note that the normalization factor $N(\mathbf{x}_i)$ in equation (10-1) ensures that $|\mathbf{v}_i(t)| \leq 1$. The vector length $|\mathbf{v}_i(t)|$ can be interpreted as the ‘velocity’ of \mathbf{x}_i at time t .

The remaining problem is to define the complete distance covered by each document representative, i.e. the total length of the path the document representatives are moved about. Since the length of $\mathbf{v}_i(t)$ depends on the single ϕ_{ij} all vectors $\mathbf{v}_i(t)$ can be scaled by a constant factor λ which is called ‘standard stepsize’. As a result, the distance covered by each individual \mathbf{x}_i will be a fraction of the standard stepsize. This fraction depends on information provided by the attractor strength matrix. What is a good value for the standard stepsize λ ? Quite clearly, λ should depend on some property of the given semantic document space X . Such a property is the average document distance as given by

$$\bar{d} =_{\text{def}} \frac{1}{n^2 - n} \cdot \sum_{i,j=1..n} |\mathbf{x}_i - \mathbf{x}_j|. \quad (10-2)$$

For a total of t_{\max} iteration steps and a parameter $p \in \mathbb{R}^+$, which is called *movement length factor*, the standard stepsize λ of the movement is defined as

$$\lambda =_{\text{def}} p \cdot \frac{\bar{d}}{t_{\max}}. \quad (10-3)$$

Finally, for $1 \leq t \leq t_{\max}$ the actual document movement is a vector translation defined by

$$\mathbf{x}_i(t+1) =_{\text{def}} \mathbf{x}_i(t) + \lambda \cdot \mathbf{v}_i(t). \quad (10-4)$$

Reconsidering the velocity metaphor sketched in 10.3.1, the ‘movement vector’ $\mathbf{v}_i(t)$ contains direction information for the movement of \mathbf{x}_i which depends on time t . The length of $\mathbf{v}_i(t)$ depends on Φ and can be interpreted as the ‘velocity’ of \mathbf{x}_i . The time for which the \mathbf{x}_i are moved is the same for all document representatives. What is the length of the longest path a document representative may take in the semantic document space? After t_{\max} movement steps the resulting position of $\mathbf{x}_i(1) =_{\text{def}} \mathbf{x}_i \in X$ is $\mathbf{x}_i(t_{\max}+1)$. According to (10-4)

$$\mathbf{x}_i(t_{\max} + 1) = \mathbf{x}_i(1) + \sum_{t=1}^{t_{\max}} \lambda \cdot \mathbf{v}_i(t). \quad (10-5)$$

Because $|\mathbf{v}_i(t)| \leq 1$ the length of the total movement path for \mathbf{x}_i can be estimated by

$$\sum_{t=1}^{t_{\max}} |\lambda \cdot \mathbf{v}_i(t)| \leq \sum_{t=1}^{t_{\max}} \lambda = t_{\max} \cdot \lambda = p \cdot \bar{d} \quad (10-6)$$

The resulting document space $X' =_{\text{def}} \{\mathbf{x}_1(t_{\max}+1), \dots, \mathbf{x}_n(t_{\max}+1)\}$ contains document representatives which have been ‘moved’ according to their ‘degree of relatedness’ as defined by the attractor-strength matrix Φ . Obviously, the final arrangement of documents in the new semantic document space X' depends on the corresponding arrangement of documents in the given semantic document space X . The movement length factor p can be used to adjust the degree to which X' reflects the structural changes as defined by Φ . The number of iterations t_{\max} controls the ‘fineness’ of the movement steps and the faithfulness to which the document representatives follow the movements of each other. Algorithm 10-1 summarizes the movement of document representatives in the semantic document space.

10.3.3 Illustration of Object Movement Defined by Attractors

Having presented the method for object movement based on vector translation, this section discusses some effects of executing attractors. Figure 10-4 depicts the movement of six objects in a two-dimensional space. Consider objects 4, 5 and 6 first: Object 6 is not attracted by any object and remains in place whereas it strongly attracts object 4. However, there is also a slight bi-directional attraction between objects 4 and 5. Note that object 4 is only attracted by objects 5 and 6, so the effect of the movement can be studied locally. As a result of the attractor execution, 4 has moved strongly towards 6 but has been deflected slightly by the less strong attraction of 5. Following the ‘velocity’ metaphor, 4 moves fast into the resulting direction as defined by the strong ‘force’ of 6 and the relatively small ‘force’ of 5. Object 5 moves towards 4 slowly and can cover only a small distance in the constant amount of time available.

A more interesting situation can be discovered for objects 1, 2 and 3: Object 3 attracts 1 moderately but is very strongly attracted by 4 itself. Also, 4 attracts 1 moderately. Finally, object 3 attracts object 2 to a medium degree. After the attractors have been executed, 1 is closer to both, 3 and 4 as one would expect. However, the distance between 2 and 3 is larger than in the initial arrangement though there has been even an attraction by 3. Considering the ‘velocity’ metaphor again this behavior becomes clear: Object 2 is not fast enough to follow 3 on its way. However, it followed 3 to some extent, thus lowering the gap that would have been opened otherwise. Indeed, such an effect is desired in the intended application field as the following intellectual experiment makes plausible: The idea of the attractor approach is to move documents in a pre-structured space according to views which are expressed in a rule base. Such a view may establish a strong topical connection between documents 3 and 4 and a weaker connection between 2 and 3. The ratio of the strengths of these connections should be reflected in the resulting structuring. This means, however, that initial similarities may be weakened because the final distances depend on both, the view and the a priori defined semantic similarities. The final constellation of documents 2, 3 and 4 reflects the high original similarity and the moderate attraction of 2 by 3 but also the strong relatedness of 3 and 4 as expressed by the rules that defined the actual attractor.

Algorithm 10-1: Document Movement in the Semantic Document Space

```

ExecuteAttractors ( $t_{\max}$ ,  $p$ ,  $\Phi$ ,  $X = \{\mathbf{x}_1(1), \dots, \mathbf{x}_n(1)\}$ )
BEGIN
    calculate  $\lambda$  based on  $t_{\max}$  and  $p$  according to equation (10-3);
    FOR  $t := 1..t_{\max}$  DO
        FOR each document representative  $\mathbf{x}_i$  DO
            compute  $\mathbf{x}_i(t+1)$  according to equation (10-4);
    return  $\{\mathbf{x}_1(t_{\max}+1), \dots, \mathbf{x}_n(t_{\max}+1)\}$ ;
END

```

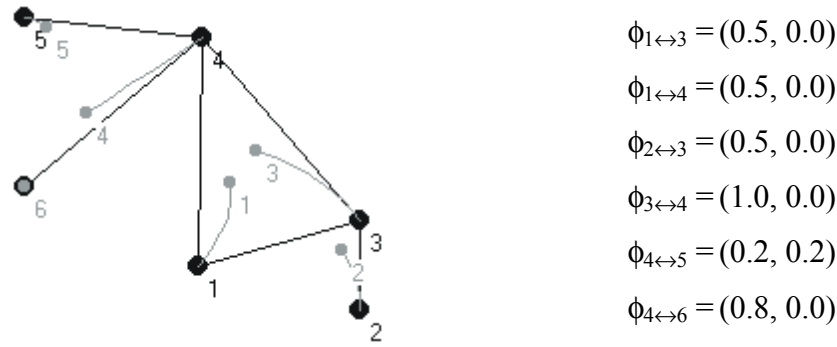


Figure 10-4: Illustration of attractor execution with $t_{\max} = 5$ and $p = 0.5$. Black points represent the initial arrangement. Black lines indicate attractors $\phi_{i \leftrightarrow j} = (\phi_{ij}, \phi_{ji})$ as defined by the attractor strength matrix Φ . Grey points symbolize the position of objects after executing the attractors. Grey lines show the movement paths.

Another interesting movement situation (already known from figure 10-3) is sketched in figure 10-5 (a): Object 2 attracts objects 1 and 3 and is attracted by object 4 itself. Objects 1 and 3 follow the attracting object 2 on its path towards 4. As a consequence, not only the distance between 2 and 4 as well as 1, 2 and 3 is reduced but necessarily also the distance between 1, 3 and 4. This shows the implicit ‘transitivity’ of attractors. Transferring this situation to document relatedness and the distance-similarity analogy of the document map, documents 1 and 3 have been ‘made more similar’ to 4 although there is no explicitly defined rule that defines an attractor between those documents. Again, this is a desired effect as the following line of thought shows: If document 2 has a topical relation to document 4 and a different thematic connection to both, 1 and 3 (a relation which does not explicitly hold between 1 and 4 or 3 and 4, respectively), then one can say that documents 1 and 3 are related to 4 by the ‘thematic bridge’ realized by document 2. However, this is a matter of interpretation. This situation is similar to the problem of whether document relationships can be expressed by metrical distance functions for which the triangular inequality hold (cf. chapter 6.2.1.2). After all, the transitive behavior is consistent with the distance metaphor of document maps.

Finally, consider figure 10-5 (b). In this case objects 7 and 8 are attracted by objects 1, 2 and 3 to some degree and therefore move towards them. They come to a halt near the ‘passive’ cluster of objects 4, 5 and 6. Following the metaphor of the document map, the corresponding documents 7 and 8 are now more similar (or related) to 4, 5 and 6 although obviously no rule

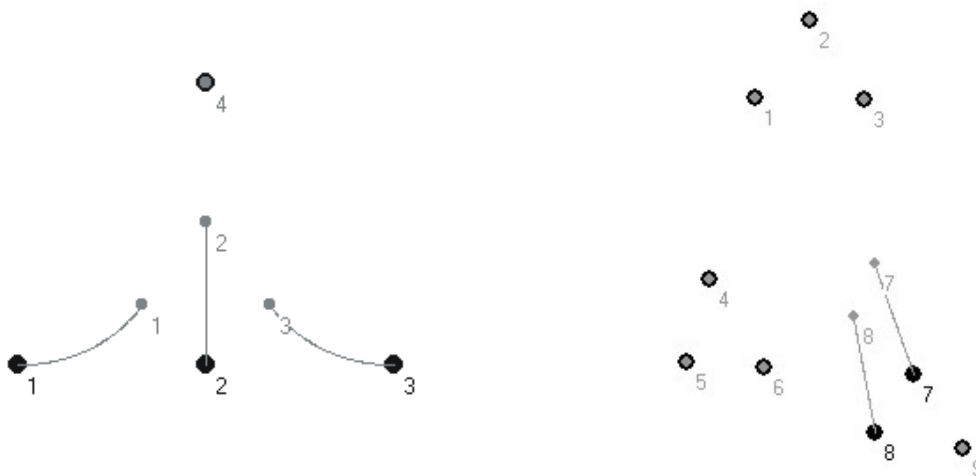


Figure 10-5: (a) Transitivity of attractor movement: Object 4 attracts 2, object 2 attracts 1 and 3, which causes also a movement of 1 and 3 towards 4. (b) Attraction of objects 7 and 8 by the cluster of objects 1, 2, 3. As a result, 7 and 8 are closer to the ‘passive’ cluster 4, 5, 6.

has been triggered that establishes a relationship between them (because there are no corresponding attractors). This situation is similar to that described just above and must also be interpreted taking into consideration the pre-structuring and the view as expressed in the rule base: As 4,5 and 6 lie nearly in between the clusters 1,2,3 and 7,8,9 there must be a defined semantic relationship between 4,5,6 to both, 1,2,3 and 7,8,9 according to the pre-structuring. The new structure which indicates also a relationship between 7, 8 and the cluster 4,5,6 must be interpreted in the light of defined rules *and* pre-structuring.

Summing up, the effects of the movement can intuitively be understood by the velocity metaphor. The interpretation of the resulting structures depends on both, the a priori defined structure of the given semantic document space and the attractors that have been defined by the rule base. The approach makes it possible to ‘tear apart’ the pre-structuring to some degree in order to reflect the imposed ‘view’ on a collection of documents. However, the resulting structure is not independent of the pre-structuring – a circumstance that is desired by the approach proposed in this section.

10.4 Overview of the Rule Design

In order to form a bias towards certain key topics and relationships between topics, a rule-based approach is proposed in this work. Rules shall define these relationships and cause an attraction between those pairs of documents in the semantic document space which meet the requirements defined in the rules. Whether a rule applies for a given pair of documents is determined by the semantic refinement module.

This section outlines the basic modeling elements and the rule types for the proposed semantic refinement module. A formal definition of their syntax and semantics will be given later. In each case the motivation leading to the introduction is discussed. There are two types of modeling elements: The basic elements are the anchors for the semantic rules which express document relationships. The rules themselves differ in their complexity from a simple weighting of certain document features to a more complex matching of feature patterns.

Note that it is not the intention to have a (in some sense) complete set of modeling elements. The focus of this work is rather on developing a method which allows a task and domain dependent influence on computed structures. Though the proposed modeling elements will be powerful enough to express interesting aspects of document relationships, extensions are possible and – from a knowledge representation point of view – desirable. The formal rule semantics will be defined in a modular fashion so that additional modeling elements can be incorporated fairly easy if that should be necessary for a special application field. The following sections present the set of modeling elements realized in this approach.

10.4.1 Basic Modeling Elements

The following modeling elements are the basis for defining the rules that finally create the actual attractors on activation. In this work two different atomic modeling elements are proposed: extended domain specific concepts – which model ‘lexical fields’ – and attribute-value pairs which are interesting especially in a technical application domain.

10.4.1.1 Extended Domain Specific Concepts

In this work concepts are defined as sets of terms which are related in a given application domain and/or analysis task. A term, on the other hand, is simply a string to be found in a docu-

ment. Concepts are defined extensionally rather than intensionally, i.e. there is no formal description of a class of objects which share certain properties. Rather, the set of terms belonging to a concept is explicitly given.

The idea of this general modeling element is that a concept defines a ‘lexical field’. From a linguistic point of view a ‘lexical field’ is a set of semantically related terms. The particular semantic structure of the field is induced by syntagmatic or paradigmatic sense-relations [Lyon81]. Syntagmatic relations hold among elements that can occur in combination with each other. Paradigmatic relations hold among intersubstitutable elements. Examples for lexical fields induced by paradigmatic semantic relations are synonyms (absolute or partial equivalence in meaning), hyponyms and hypernyms (sub-concepts and super-concepts), or antonymic terms (terms with contrary meanings).

In the application domain of knowledge management the perception of which terms belong to a lexical field may be highly context-dependent. In this sense a concept can be seen simply as a grouping of terms which should not be distinguished in the context of the current analysis. Besides, it is easy to see that relatedness of meaning is clearly a matter of degree (cf. [Lyon81]): A term might fit into the grouping under consideration only to a certain degree. Again, the degree of membership depends on the application domain as well as the given analysis task. In the approach proposed here concepts are sets of *weighted* terms. The weights indicate the degree to which a term is related to the concept.

A simple example shall clarify these ideas: The extension of the domain specific concept ‘companies’ may be a set of relevant company names which can occur in interesting documents. In an information technology context the set may include ‘IBM’, ‘Microsoft’, ‘SAP’ or ‘Oracle’:

```
companies := <IBM, 1.0, Microsoft, 1.0, SAP, 1.0, Oracle, 1.0>
```

An analyst may want to define a concept ‘strong competitors’ which comprises all companies which compete with the analyst’s firm in a certain field. Depending on the market sector under consideration the single companies may be more or less strong competitors:

```
strong-competitors := <Microsoft, 0.6, SAP, 1.0, Oracle, 1.0>
```

Whereas the concept ‘companies’ contains all terms to the highest degree, only the terms ‘SAP’ and ‘Oracle’ qualify as full ‘strong competitors’. ‘Microsoft’ belongs to the field only to a certain degree.

Modeling domain specific concepts as sets of weighted terms leads to the notion of fuzzy sets (cf. section 10.5.1). Numerically encoding imprecise meaning of terms by fuzzy sets is not a new idea but has been studied and theoretically underpinned by linguists. Imprecision in the sense of text linguistics includes both, *variants* of meaning due to individual, situative, social, educative or historic influences as well as formal *vagueness* which is characteristic for symbol systems of natural languages [Rieg90]. In the notion of linguistics an extensional definition of a fuzzy set for describing a concept constitutes referential semantics. In order to obtain formal referential representations of semantics introspection is normally applied (as in our case) or test persons and experts are questioned. Rieger [Rieg89] extends the referential approach by a structural modeling technique which is empirical and procedural and thus does not depend on subjective introspection (note, however, that in our context this subjective assessment is exactly intended). A similar approach is introduced in [Her98] where lexical fields are automatically computed by a context analysis of certain keywords. By measuring the contextual

‘closeness’ of terms to the keyword a degree of affinity is determined⁷. For a detailed discussion of semantic vagueness, its role and history in linguistics, its empirical analysis and formal representation see [Rieg89].

10.4.1.2 Attribute-Value Pairs

Pairs of attributes and their values play an important role especially in the technical domain. For example, the state of technical systems can frequently be expressed by the current value of certain attributes. In technical documents attribute-value pairs frequently describe important features of a system the document deals with. These features may be of a particular interest in a given analysis task.

As an example consider a product description of a commercial airplane. The following text is an excerpt from a description of the MD-80 (available at www.boeing.com):

Four MD-80 models – the MD-81, MD-82, MD-83, and MD-88 – are 147 feet, 10 inches (45.08 meters) long and accommodate a maximum of 172 passengers. The MD-87 is 130.4 feet (39.76 m) in length, with a maximum passenger capacity of 139. Wingspan for all models is 107 feet, 10 inches (32.88 m).

In this text three attributes can be found: length, passenger capacity and wingspan. Clearly, these are important features of the system described in the text fragment. Suppose that an analyst has to perform a knowledge management task that requires, besides the pre-analysis as performed by a suitable document analysis module, a certain bias towards aircraft with high wingspan. Thus, modeling attribute-value pairs is an important aspect of a semantic refinement module. Moreover, the example motivates another requirement: The actual value of a (technical) attribute which can be extracted from a document may be a crisp numeric value (“wingspan is 32.88 m”). Often, the exact value is not relevant as long as it falls into a range which can be described by a more or less vague interval. These intervals can be denoted by adjectives like ‘high’ or ‘medium’. Their interpretation depends on the attribute for which it describes the value, and, of course, on the application domain.

Attribute-value pairs consist of a noun described by an adjective or numeral. In this work attributes are concepts (linguistically usually lexical fields of nouns) which can be associated with a certain value. The value itself must be describable on a numerical scale. It will be described by a (vague) interval or a real valued number. Attribute-value pairs form the second basic modeling element proposed in this work.

10.4.2 Rule Types

Rules combine and associate the basic modeling elements that have been sketched above. In other words, they use concepts or attribute-value pairs to describe semantic relationships between documents. In the following subsections some possible rule types are motivated and sketched (without claiming completeness).

10.4.2.1 Weighting Concepts and Attribute-Value Pairs

The simplest way to add semantic information or to express views on the collection is to stress important topics which can roughly be described by certain key concepts or attribute-value pairs. Since we deal with specialized text collection this seems appropriate: A special-

⁷ The resulting lexical fields can be graphically presented as stars where the context words are circularly arranged around the keyword. The distance of each satellite to the keyword reflects the degree of affinity.

ized field of interest usually has key topics that are represented by certain key concepts. In the simplest form, the occurrence of these key elements in two documents establishes a relationship between them which should result in an attraction of the corresponding document representatives in the semantic document space.

Consider, for example, an analyst who wants to define, among other issues, a bias towards documents which contain the task- and domain-dependent key concept ‘strong competitors’. Or consider a user from a technical domain, e.g. an airplane manufacturer, who needs to analyze document collections regarding technical parameters, such as aircraft with medium wingspan and large passenger capacity. As discussed in section 10.4.1.2, these technical parameters can be described by attribute-value pairs.

To support such a simple weighting (which is supposed to be an important and useful rule class in practice) an appropriate rule type should be defined: Taking up the examples again, weighting-rules as proposed in this work look as follows:

```
<strong-competitors> 0.7
    <medium wingspan> 1.0
    <large passenger-capacity> 0.5
```

Each rule consists of a concept or attribute-value pair and a weight factor. To give an informal presentation of the rule semantics: Given two documents, an attraction between them is set up if both documents match the concept or attribute-value pair. The strength of the attraction depends on the strength of the matching and the weight factor. Thus, the weight factor can be used to form priorities for the bias. For example, consider the last two rules in the application domain ‘airplanes’. Here, the rules define a strong connection between documents about aircraft with medium wingspan but only a medium attraction between documents about large passenger capacity (indicated by the weights).

10.4.2.2 Patterns of Basic Modeling Elements

Often it is necessary to express a relationship between concepts in a more differentiated way than treating them as equal. Patterns of basic modeling elements provide a means to express asymmetric relationships between different sets of modeling elements. As an example consider the application domain ‘Internet protocols’, in particular ‘telnet’ and ‘hostname’ protocols. Assume that in a certain analysis task the ‘hostname’ concept is of a particular importance. Moreover, it may be desired that all documents which are related in some sense to the hostname protocol, e.g. texts about the ‘telnet’ protocol, “move” closer to ‘hostname’ documents. In this way a certain bias towards ‘hostname’ protocols is formed. The rationale behind this is that, since in a telnet session the client needs to address the host computer by its name, the concept ‘telnet’ is related to ‘hostname’. However, the ‘hostname’ protocol is somehow more general than the ‘telnet’ concept.

This asymmetric relationship can be expressed by a rule which consists of two rule sides: One side contains the concepts or attribute-value pairs that have to be found in one document, the other side contains a (possibly different) list of concepts or attribute-value pairs that have to be found in another document. As a consequence, an attraction between documents that match the pattern is set up. Again, the degree of rule triggering can be adjusted by a user-defined weight factor.

As an example consider the rule

```
<telnet> 0.8 <hostname> 0.2
```

which is interpreted as follows: Documents containing the concept ‘telnet’ will be attracted by ‘hostname’ documents with a relative strength of 0.8, and texts about ‘hostname’ will be attracted weakly (with a relative strength of 0.2) by ‘telnet’ documents. Again, the absolute attractive force between each pair of documents also depends on the matching degree of both concepts or attribute-value pairs.

10.4.2.3 Syntactic Concept Patterns

The rule types sketched so far are document-oriented: A rule triggers if all modeling elements of both sides of the rule can be found in the respective document under consideration, regardless if the concepts are grammatically related. While this is useful for some tasks it is not always desired. Assume that an analyst wants to establish a connection between documents (e.g. Internet protocols) about ‘sending e-mail’ and ‘receiving e-mail’. A rule

`<send ,e-mail> 1.0 <receive, e-mail> 1.0`

of the rule type discussed above would not do the job since it triggers if only the concepts ‘send’ and ‘e-mail’ appear somewhere in one document and ‘receive’ and ‘e-mail’ appear somewhere in the other. Furthermore, it is usually useful to match terms by reducing them to their *stems* (cf. section 6.2.1.4). Consequently, a term ‘send’ would not only match the terms ‘send’ or ‘sends’ but also the terms ‘sender’ or ‘sending’.

For this reason a more elaborated rule type is proposed: Syntactic concept patterns allow the combination of concept patterns with syntax information. Again, these rules have two sides which define a pattern to be matched against one document each.

As an example, consider the modified e-mail rule

`<send/VB, e-mail/NN> 1.0 <receive/VB, e-mail/NN> 1.0`

which defines that the concept ‘send’ or ‘receive’, respectively, has to be a verb (‘VB’) and ‘e-mail’ a noun (‘NN’). Rules of this type trigger only if there is a sentence in the first document which contains ‘send’ as a verb and ‘e-mail’ as a noun and there is a sentence in the second document which contains ‘receive’ as a verb and ‘e-mail’ as a noun. In both cases, verb and noun should be grammatically related.

10.5 Excursus: Techniques for Implementing Rules

This section briefly sketches some basic techniques used for the rule-based approach. Section 10.4.1 has shown that we encounter some vagueness when it comes to dealing with concepts and attribute-value pairs. Vague information can be processed using methods of fuzzy set theory or fuzzy logic (cf. section 10.5.1). Fuzzy set theory will be incorporated directly into the semantics of the basic modeling elements. In order to realize the rule type of syntactic concept patterns (section 10.4.2.3) it is necessary to obtain syntactic information about the category of words in texts. This can be achieved by applying methods of shallow natural language processing, e.g. part of speech tagging (section 10.5.2). Since tagging will only be used for preprocessing documents the actual method applied in a realization is not significant. However, in order to provide some background, the method chosen in the realization of this work is briefly presented.

10.5.1 Fuzzy Set Theory

Fuzzy set theory is a generalization of classic set theory, originally developed by Lotfi Zadeh [Zad65]. It is a way of processing vague data by allowing a partial set membership of elements rather than requiring a crisp ‘to belong to or not to belong to’. The rationale behind this idea is that humans do not require precise, numerical information, yet they are capable to effectively control complex systems by dealing with imprecise, vague information. Originally, fuzzy based systems were successfully used for control applications since fuzzy technology tries to mimic human control logic [Zad87]. Meanwhile, fuzzy set theory and fuzzy logic are effectively used in various areas, e.g. for supporting man-machine communication [BoPi95, TBK+96].

Formally, a *fuzzy set* F over universe U is a mapping $F: U \rightarrow [0,1]$. F is also called membership function, and $F(u)$ is called the degree of membership of $u \in U$ in F . The membership function can be seen as an extension of the classic characteristic function χ of a crisp set X where $\chi: X \rightarrow \{0,1\}$ with $\chi(x) = 1 \Leftrightarrow x \in X$. Some authors distinguish the membership function itself (often denoted as μ_F) and the fuzzy set $F =_{\text{def}} \{(u, \mu_F(u)) \mid u \in U\}$. Since this differentiation is useless this work treats the notions ‘fuzzy set’ and ‘membership function’ as equal.

The classic operations on sets – union, intersection and complement – can be directly transferred to fuzzy sets: Just as classic set operations can be derived from connectives of predicate logic (disjunction, conjunction and negation), fuzzy set operations are based on disjunction, conjunction and negation in fuzzy logic. Roughly spoken, these operations are extensions of the two-valued logical connectives to the standard interval $[0,1]$ (so-called Lukasiewicz extensions of Boolean functions, possibly with additional axioms). There are many realizations (even parameterized classes) of Boolean connectives and operations in fuzzy logic and fuzzy set theory (e.g. [Yag80, DuPr80, DuPr85, KlFo88]). Most popular are the following definitions: For fuzzy sets $F, G: U \rightarrow [0,1]$, union, intersection and complement are defined based on minimum-conjunction, maximum-disjunction, and Lukasiewicz-negation:

$$(F \cup G)(u) =_{\text{def}} \max(F(u), G(u)),$$

$$(F \cap G)(u) =_{\text{def}} \min(F(u), G(u)),$$

$$\bar{F}(u) =_{\text{def}} 1 - F(u).$$

In applications of fuzzy set theory the notion of linguistic terms and linguistic variables can be found often. A *linguistic term* is a natural language identifier, mostly an adverb or adjective.

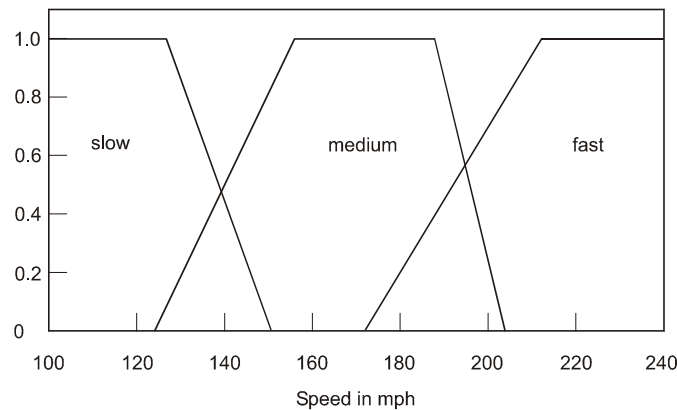


Figure 10-6: Fuzzy sets over domain ‘speed in mph’. The names ‘slow’, ‘medium’ and ‘fast’ are linguistic terms, a suitable linguistic variable could be called ‘speed’.

tive and as such a purely linguistic construct. To interpret these identifiers, a fuzzy set is assigned to each linguistic term as an operational representation (in this sense the fuzzy set is the formal semantics of a linguistic term). A *linguistic variable* is a variable which can take linguistic terms as its values. Of course, it is necessary to specify a domain for the linguistic variable and permissible vague values (named by linguistic terms). Figure 10-6 sketches some fuzzy sets over a continuous numerical domain.

10.5.2 Part of Speech Tagging

Part of speech tagging is the process of annotating natural language texts with syntactic, linguistic information depending on the context the words appear in – an important step for any kind of natural language processing and understanding. Due to the ambiguity of natural languages and the flexibility of grammar, part of speech tagging is a hard task. According to [XBC94] tagging methodology can be broadly categorized as qualitative and quantitative methods: Qualitative approaches deterministically decide a word's part of speech by using grammatical and lexical rules that are matched against the context of the considered word. Quantitative approaches are statistical analysis methods which determine a word's part of speech by considering the overall likelihood of a set of random variables with associated probabilities.

In recent years quantitative methods have been moved into the focus of research again since they have some interesting properties (cf. [LaNa95]): Among others, they are robust since they are capable of dealing with noisy input, they are efficient and effective and avoid the bottleneck of linguistic knowledge representation. On the other hand, these methods have the disadvantage that linguistic knowledge is only captured indirectly in large tables of statistics [Bri94]. A different approach is Eric Brill's transformation-based tagger that uses a trainable, rule-based system to store the linguistic information needed for tagging [Bri92, Bri94]. It achieves performance comparable to that of stochastic taggers and its implementation is publicly available.

The tagger is based upon a learning paradigm called *transformation-based error-driven learning* [Bri94]. First, the un-annotated text is taken and each word is assigned its most likely tag as indicated in a training corpus. For unknown words the tag is guessed based on a number of linguistic features, such as whether the word is capitalized, and what the last three letters of the word are. The learning algorithm follows a greedy strategy: To learn a transformation rule, every possible instantiation of pre-defined transformation templates is applied to the sentence and the number of tagging errors is determined by comparing the result to the *truth* which is defined by a manually tagged training corpus. The transformation resulting in the greatest error reduction is then chosen and added to the rule-base. Once the system is trained, new sentences are tagged by feeding the text to the initial state annotator and then applying each transformation, in turn, to the sentence.

The tags assigned by the tagger follow the *Penn Treebank Part of Speech Tags* [Bri95], shown in appendix E. The following example presents a sentence taken from the US patent database before and after annotation with part of speech tags:

The structure and method allow the locating of a stored record in a massive system in a controlled and small number of mass memory accesses.

The/DT structure/NN and/CC method/NN allow/VB the/DT locating/VBG of/IN a/DT stored/VBN record/NN in/IN a/DT mas-

sive/JJ system/NN in/IN a/DT controlled/VBN and/CC small/JJ
number/NN of/IN mass/NN memory/NN accesses/NNS ./.

10.6 Realization of the Proposed Rule Base Approach

Having informally introduced different rule types and basic modeling elements for defining a bias towards an analyst's interest (section 10.4), the proposed rule base approach is now specified formally. Section 10.6.1 presents the formal semantics of the basic modeling elements before section 10.6.2 defines functions that match patterns of basic modeling elements against documents. Finally, section 10.6.3 formalizes the rule types as motivated above and defines functions that match a rule against documents.

In order to properly define the basic modeling elements and rule types along with their semantics, an abstract view on the document collection is required. The semantic refinement module introduced in this chapter assumes that a document is represented as an ordered set of sentences. Each sentence consists of terms that have been annotated with syntactic information by a part of speech tagger (cf. section 10.5.2).

10.6.1 Concepts and Attribute-Value Pairs

The *concept* is the basic knowledge modeling element in this work. Its main function is to serve as a domain specific 'lexical field' which combines terms that should not be distinguished in the given analysis task (cf. section 10.4.1.1). Let Σ denote an alphabet. A *term* t is an element of $T \subseteq \Sigma^*$. T is the vocabulary containing all possible terms.

Formally, a concept c is a fuzzy set of terms, i.e. $c: T \rightarrow [0,1]$ where $c(t)$ is the degree of membership of term t in concept c . Concepts are defined extensionally, i.e. the fuzzy set is given explicitly by enumerating its elements t with $c(t) > 0$. Let C denote the set of all concepts.

As an example consider the definition of a concept 'strong competitors' already known from above:

`strong-competitors := <Microsoft, 0.6, SAP, 1.0, Oracle, 1.0>`

Here, the formal semantics of this definition is a fuzzy set $c_{\text{strong-competitors}}$ with

$c_{\text{strong-competitors}}(\text{Microsoft}) = 0.6$, $c_{\text{strong-competitors}}(\text{IBM}) = 1$, $c_{\text{strong-competitors}}(\text{Oracle}) = 1$,

and $c_{\text{strong-competitors}}(t) = 0$ for all $t \in T \setminus \{\text{Microsoft, IBM, Oracle}\}$.

A special kind of concept is an attribute. *Attributes* are concepts which represent a domain-specific numerical scale (like bandwidth, height or wingspan). Their *values* can be described by numerals or suitable adjectives (like low, medium or high). Formally, an attribute a is a triple $a = (c, G[c], V)$ where $c \in C$ is a concept, $G[c]$ its domain, and V a set of linguistic terms that describe possible vague value intervals. Each $v \in V$ is implicitly associated with a concept c_v where $c_v(v) = 1$ and $c_v(t) = 0$ for all terms $t \neq v$. An attribute a can be seen as a linguistic variable with name c (more precisely, each term t with $c(t) > 0$ is a suitable name).

A linguistic term $v \in V$ is interpreted by a trapezoid fuzzy set $F_v: G[c] \rightarrow [0,1]$ over a domain $G[c] = [g_{\min}, g_{\max}] \subset \mathbb{R}$ that defines the range of values that the concept c can be described with. For defining such a fuzzy set the user gives four values z_1, z_2, z_3 and z_4 , $z_1 \leq z_2 \leq z_3 \leq z_4$, with $z_1, z_4 \in \mathbb{R}$ and $z_2, z_3 \in G[c]$. Then, the fuzzy set $F_v: G[c] \rightarrow [0,1]$ is defined as

$$F_v(x) =_{\text{def}} \begin{cases} \frac{1}{z_2 - z_1} \cdot x - \frac{z_1}{z_2 - z_1} & \text{for } z_1 \leq x < z_2 \\ 1 & \text{for } z_2 \leq x \leq z_3 \\ -\frac{1}{z_4 - z_3} \cdot x + \frac{z_4}{z_4 - z_3} & \text{for } z_3 \leq x < z_4 \\ 0 & \text{otherwise} \end{cases}, \quad (10-7)$$

for all $x \in G[c]$. Figure 10-7 depicts some trapezoid fuzzy sets defined by the four base points z_i . Note that the relaxation that z_1, z_4 do not necessarily have to belong to $G[c]$ allows the definition of fuzzy sets like F_{low} and F_{high} from figure 10-7. If $z_1 < z_2 = z_3 < z_4$ the fuzzy set F_v represents a vague number, e.g. ‘approximately 100’. If $z_1 = z_2 = z_3 = z_4$ the fuzzy set F_v represents a singleton. Such fuzzy sets model crisp numbers, e.g. $z_i = 100$ or $z_i = 1.2$, etc. The definition above is limited to trapezoid functions for efficiency reasons (more precisely, the limitation allows an efficient matching of different fuzzy sets). Theoretically, of course, other functions would be possible as well.

Finally, for a given attribute $a = (c, G[c], V)$ the set Ω_a contains all interpretations of linguistic terms of attribute values for c and all singleton fuzzy sets for crisp values (numbers) $z \in G[c]$, i.e. $\Omega_a =_{\text{def}} \{F_v \mid v \in V\} \cup \{F_z \mid z \in G[c] \text{ and } z = z_i, i = 1, \dots, 4\}$. For a given attribute $a = (c, G[c], V)$ an *attribute-value pair* is a tuple (c, v) where $v \in V$.

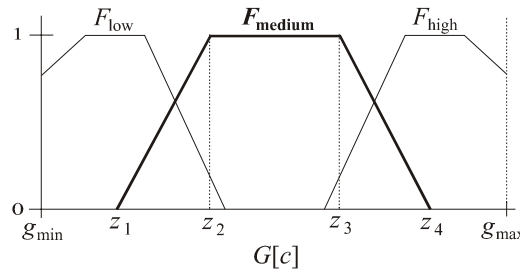


Figure 10-7: Trapezoids fuzzy sets over domain $G[c]$ defined by base points z_1, z_2, z_3 and z_4

The (vague) values $F_v \in \Omega_a$ of an attribute $a = (c, G[c], V)$ can be compared against each other by the fuzzy relation $\text{sub}: \Omega \times \Omega \rightarrow [0, 1]$ (where Ω denotes the set of all trapezoid fuzzy sets) which is an asymmetric relationship between fuzzy sets F_v and $F_{v'} \in \Omega_a$.

Let $\int F$ denote $\int_{z_1}^{z_4} F(x) dx$. Then, $\text{sub}: \Omega \times \Omega \rightarrow [0, 1]$ is defined as

$$\text{sub}(F_v, F_{v'}) =_{\text{def}} \begin{cases} \frac{\int (F_v \cap F_{v'})}{\int F_v} & \text{for } \int F_v > 0 \\ F_{v'}(F_v^{-1}(1)) & \text{for } \int F_v = 0 \end{cases} \quad (10-8)$$

The case $\int F = 0$ represents a singleton fuzzy set, i.e. $z_1 = z_2 = z_3 = z_4$ and $F_v^{-1}(1) = z_i$.

The idea underlying this definition is to model a soft asymmetric matching relation for fuzzy sets, allowing statements of the form ‘ F_v covers $F_{v'}$ to a certain degree’. Figure 10-8 illustrates the fuzzy relationship $\text{sub}: \Omega \times \Omega \rightarrow [0, 1]$ for fuzzy sets F_v and $F_{v'}$. The first figure represents two disjoint sets, the second a total inclusion of $F_{v'}$ in F_v where the relationship $\text{sub}(F_v, F_{v'})$ still holds to some degree, since $F_{v'}$ covers a good portion of F_v . The third figure sketches a partial matching where $F_{v'}$ covers a larger portion of F_v than vice versa.

Note that the relation *sub*, which resembles a fuzzy inclusion relation, could also have been defined on a stricter logical basis: A *classic* set inclusion $X \subseteq Y$ over a domain U is logically defined as an implication $\forall u \in U: u \in X \Rightarrow u \in Y$ where $A \Rightarrow B$ is equivalent to $\neg A \vee B$ for truth values A and B . Transferred to fuzzy sets and fuzzy logic, a fuzzy inclusion $inc(F_v, F_{v'})$ of fuzzy sets F_v and $F_{v'}$ over domain U is defined as $inc(F_v, F_{v'}) =_{def} \inf \{ \text{imp}(F_v(x), F_{v'}(x)) \mid x \in U \}$ with $\text{imp}(F_v(x), F_{v'}(x)) =_{def} \max(1 - F_v(x), F_{v'}(x))$ (cf. section 10.5.1). For trapezoid fuzzy sets the degree of inclusion can be computed efficiently. However, for a degree of matching greater zero the definition requires that each x with $F_v(x) = 1$ is included in $F_{v'}$ to some degree, i.e. $\forall x: F_v(x) = 1 \Rightarrow F_{v'}(x) > 0$. Otherwise, $inc(F_v, F_{v'}) = 0$. Since this is a hard constraint for an intuitive modeling of vague attribute values (e.g. figure 10-8(c) would produce a zero matching, though intuitively there is a certain matching) the more pragmatic approach of definition (10-8) has been taken.

10.6.2 Matching Modeling Elements Against Documents

Having introduced the basic modeling elements, prior to defining the rules and their semantics, it is necessary to discuss the question whether a given concept c or an attribute-value pair (c, v) matches a given document d . Moreover, not only simple concepts, but more complex concept patterns with syntactic information will be examined.

At this point it is important to define the formal representation of a document for the semantic refinement module. As defined earlier, T is the set of all terms that may appear in a document. Let Θ be the set of all syntactic tags which can be assigned to terms by the part of speech tagger actually used. A *sentence* s is a sequence of tagged terms, represented as tuples $w \in T \times \Theta$, $s = w_1, \dots, w_k$, $w_i = (t_i, \theta_i)$. The set of all possible sentences is denoted by S . Finally, a *document* d is an ordered set of sentences, $d \subseteq S$, and D denotes the set of all formal document representations.

More specifically, in this section the problem of *sentence* matching will be examined. For concepts, attribute-value pairs, as well as concepts and concept patterns accompanied by additional syntactic information, a suitable matching frequency function v and matching weight function μ will be defined. The matching frequency function specifies how often a modeling

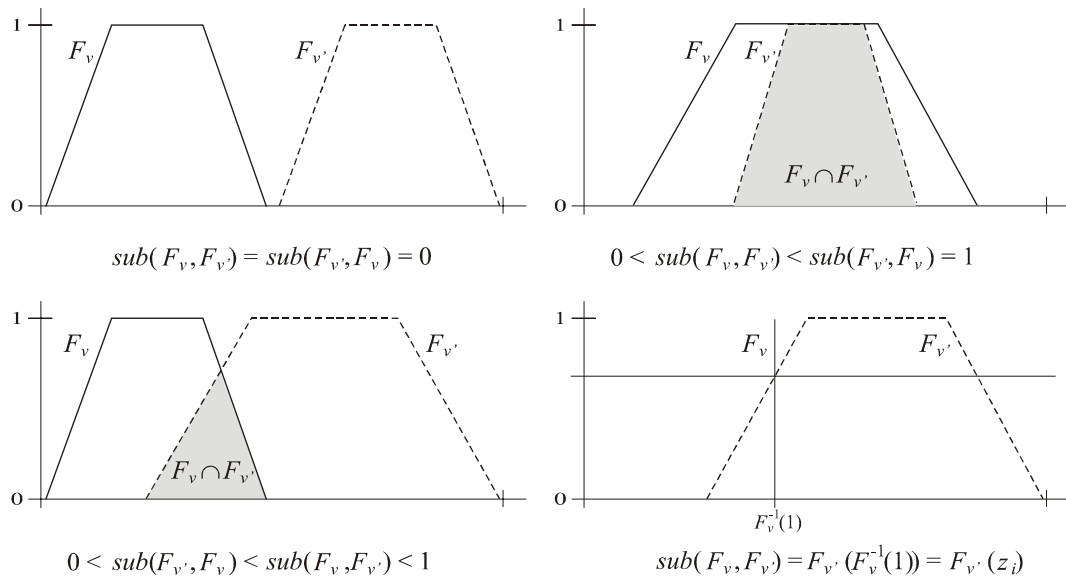


Figure 10-8: Illustration of the fuzzy relationship $sub: \Omega \times \Omega \rightarrow [0,1]$ for fuzzy sets F_v and $F_{v'}$. From left to right and top to bottom: (a) no matching, (b) total inclusion, (c) partial overlapping, (d) inclusion of a singleton

element (possibly respecting the given syntactic constraints) appears in a sentence. The matching weight function determines a degree to which the modeling element matches.

10.6.2.1 Simple Concept Matching

A simple concept (which is a fuzzy set of terms) matches a given sentence if and only if some term of the sentence is also a member (to some degree) of the concept's term set. The matching frequency is increased by one for every matching term of the sentence. The weight of the matching is determined by the degree of membership of the sentence's term in the concept's fuzzy set of terms. The matching itself neglects the actual word form of the term by only considering the root form of both, terms in the sentence and terms in the fuzzy set. For a term t let $\text{stem}(t)$ denote the word stem of term t (cf. chapter 6.2.1.4).

The matching frequency function v_c for concepts is defined as

$$v_c: C \times S \rightarrow \mathbb{N}_0 \quad (10-9)$$

$$v_c(c, s) =_{\text{def}} |\{i \mid \text{for } w_i = (t_i, \theta_i): \exists t \in T: \text{stem}(t) = \text{stem}(t_i) \wedge c(t) > 0\}|$$

where w_i denotes the i -th term×tag-pair in sentence s . Let x_i denote the degree of membership of term t_i in concept c , i.e.

$$x_i =_{\text{def}} \begin{cases} c(t) & \text{if } w_i = (t_i, \theta_i) \wedge \exists t \in T: \text{stem}(t) = \text{stem}(t_i) \\ 0 & \text{otherwise} \end{cases} \quad (10-10)$$

Then, the matching weight function μ_c for concepts is defined as

$$\mu_c: C \times S \rightarrow \mathbb{R}^+ \quad (10-11)$$

$$\mu_c(c, s) =_{\text{def}} \sum_{i=1}^k x_i.$$

10.6.2.2 Matching of Tagged Concepts

The simple concept matching neglects the grammatical category of a word in the sentence. A concept containing the term 'computer', for example, will also match a sentence containing the word 'computes'. If it is desired to limit possible matchings to certain word categories a simple constraint is to specify a part-of-speech tag for defining the concept to match more clearly. If a tag is specified, for a successful matching the tag must match the word's tag in the sentence as well. This leads to a simple variation of v_c and μ_c .

The matching frequency function $v_{c,\theta}$ for concept-tag pairs $(c, \theta) \in C \times \Theta$ is defined as

$$v_{c,\theta}: C \times \Theta \times S \rightarrow \mathbb{N}_0 \quad (10-12)$$

$$v_{c,\theta}(c, \theta, s) =_{\text{def}} |\{i \mid \text{for } w_i = (t_i, \theta_i): \theta = \theta_i \wedge \exists t \in T: \text{stem}(t) = \text{stem}(t_i) \wedge c(t) > 0\}|$$

Analogous, the definition of x_i has to be extended:

$$x_i =_{\text{def}} \begin{cases} c(t) & \text{if } w_i = (t_i, \theta_i) \wedge \theta = \theta_i \wedge \exists t \in T: \text{stem}(t) = \text{stem}(t_i) \\ 0 & \text{otherwise} \end{cases} \quad (10-13)$$

The matching weight function $\mu_{c,\theta}$ for concept-tag pairs is then defined as

$$\mu_{c,\theta}: C \times \Theta \times S \rightarrow \mathbb{R}^+ \quad (10-14)$$

$$\mu_{c,\theta}(c,\theta,s) =_{\text{def}} \sum_{i=1}^k x_i.$$

10.6.2.3 Matching of Tagged Concept Patterns

The matching of concepts as defined above is a simple form of keyword matching. Tagged concepts contain some syntactic information to restrict possible matchings, thus making the analysis process more precise. A more complex question is whether a given sentence contains a certain phrase. For example, the analyst of a collection may want to form a certain bias for documents which deal with “sending messages”. A tagged concept pattern which defines this phrase could be

$$(\text{send}, \text{VB}, \text{message}, \text{NN}),$$

where ‘VB’ is a tag that denotes a verb and ‘NN’ stands for a noun. Grammatically, a verb-noun phrase containing the concepts ‘send’ and ‘message’ has to be found. In order to properly identify such phrases a deeper analysis of a sentence would be necessary. More precisely, a linguistically correct decision whether a sentence fragment defined by a tagged concept pattern matches a given sentence would require a syntactic analysis according to natural language grammar. This is a non-trivial problem that falls into the field of natural language processing. Natural language grammars can be approximated by context-free grammars with constraints. Parsing such grammars is a very time intensive problem and does not guarantee a correct solution [LaNa95]. Since the focus of this work is on analyzing highly specialized (mostly) technical or scientific text collections which are often characterized by a clear language without too many stylistic variations a simple approach utilizing shallow natural language processing is proposed here.

Moreover, following the idea of the staged approach, there should be a simple means of defining such phrasal patterns which can be enhanced by a more elaborated definition if necessary and desired. Consequently, the matching method for tagged concept patterns comes in two flavors: In its simple form it requires only the concept-tag pattern and the given sentence and decides whether the sentence matches all concept-tag pairs. Of course, this approach is a simple heuristic which may produce linguistically false matchings. The second matching method makes use of additional grammatical constraints given by regular expressions which define some legal grammatical phrase constructs.

The simple matching frequency function for concept-tag patterns $(c_1, \theta_1, \dots, c_l, \theta_l)$ is defined as

$$\begin{aligned} v_{(c,\theta)^l}^1 : (C \times \Theta)^l \times S &\rightarrow \{0,1\} \\ v_{(c,\theta)^l}^1(c_1, \theta_1, \dots, c_l, \theta_l, s) &=_{\text{def}} \begin{cases} 1 & \text{if } \forall i = 1..l : v_{c_i, \theta_i}(c_i, \theta_i, s) > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (10-15)$$

The matching weight with which a given pattern of concept-tag pairs matches a sentence is defined as the average matching weight of the individual concept-tag pairs of the pattern. Formally, the matching weight function $\mu_{(c,\theta)^l}^1$ is given by

$$\mu_{(c,\theta)^l}^1 : (C \times \Theta)^l \times S \rightarrow \mathbb{R}^+ \quad (10-16)$$

$$\mu_{(c,\theta)^l}^1(c_1, \theta_1, \dots, c_l, \theta_l, s) =_{\text{def}} \begin{cases} \frac{\sum_{i=1}^l \mu_{c_i, \theta_i}(c_i, \theta_i, s)}{\sum_{i=1}^l v_{c_i, \theta_i}(c_i, \theta_i, s)} & \text{if } v_{(c,\theta)^l}^1(c_1, \theta_1, \dots, c_l, \theta_l, s) = 1 \\ 0 & \text{otherwise} \end{cases}$$

As this matching function is just a very coarse heuristic it will sometimes produce false matchings as the following example shows. Consider the sentences

- a) *Eva sends Adam to the post office to get her messages.*
- b) *Eva sends Adam a message.*

where ‘send’ would be tagged as a verb (VB) and ‘message’ as noun (NN).

For a given concept pattern

$$cp = (\text{send, VB, message, NN})$$

and the method described above, both sentences would return a positive matching result. However, the intention of defining the concept pattern cp was to state that sentences about ‘sending messages’ should match. Only the second sentence meets the requirement whereas the first sentence should rather match a ‘receive messages’ concept-tag pattern which is, however, very hard to detect because it requires a deep natural language understanding.

The coarse heuristic can be improved by defining additional grammatical constraints given by syntactic regular expressions. A *regular expression* is based on a context-free grammar with only left or right linear productions. It produces strings based on concatenation, union and Kleene closure. The language class which can be described by regular expressions is exactly the class recognized by finite automata.

In this work a *syntactic regular expression* is defined as follows: Let Σ_{reg} be an alphabet which contains syntactic tags and variables, i.e. $\Sigma_{\text{reg}} =_{\text{def}} \Theta \cup \Xi$, where $\Xi = \{\xi_1, \dots, \xi_l\}$, $l \in \mathbb{N}$ is a set of variable symbols (there are as many variable symbols as concept-tag pairs in the pattern to be matched). A regular expression build over Σ_{reg} is called a syntactic regular expression. Let P be the set of all possible syntactic regular expressions. For a given set $\rho \subseteq P$ of syntactic regular expressions $L_\rho \subseteq (\Sigma_{\text{reg}})^*$ denotes the language formed by ρ .

In order to define the matching frequency and matching weight function for tagged concept patterns with syntactic regular expressions two auxiliary functions are necessary. The first is a substitution function which takes a concept-tag pattern and a single term of the given sentence accompanied by its tag. The function yields a special symbol (variable) if the sentence’s term matches a concept and its tag of the given concept-tag pattern. Otherwise, it returns simply the tag of the sentence’s term:

$$\begin{aligned} \zeta: (C \times \Theta)^l \times T \times \Theta &\rightarrow \Sigma_{\text{reg}} \\ \zeta(c_1, \theta_1, \dots, c_l, \theta_l, t, \theta) &=_{\text{def}} \begin{cases} \xi_i & \text{if } \exists i: v_{c_i, \theta_i}(c_i, \theta_i, (t, \theta)) = 1 \\ \theta & \text{otherwise} \end{cases} \end{aligned} \quad (10-17)$$

Having defined the substitution of a single term-tag pair of the given sentence, a substitution of the complete sentence is a string defined by

$$\zeta^*: (C \times \Theta)^l \times S \rightarrow (\Sigma_{\text{reg}})^* \quad (10-18)$$

$$\zeta^*(c_1, \theta_1, \dots, c_l, \theta_l, s) =_{\text{def}} \prod_{i=1}^k \zeta(c_1, \theta_1, \dots, c_l, \theta_l, t_i, \theta_i)$$

The substitution function takes a sentence and a pattern of concept-tag pairs and eliminates the terms of the sentence unless they match one of the given concept-tag pairs. The variable symbols in the substituted sequence mark successful matches of the given concept-tag pattern.

Now, the enhanced matching frequency function for concept-tag patterns $(c_1, \theta_1, \dots, c_l, \theta_l)$ and a set of syntactic regular expressions $\rho \subseteq P$ is defined as

$$\begin{aligned} v_{(c, \theta)^l}^2 : (C \times \Theta)^l \times \mathcal{P}(P) \times S &\rightarrow \{0, 1\} \\ v_{(c, \theta)^l}^2(c_1, \theta_1, \dots, c_l, \theta_l, \rho, s) &=_{\text{def}} \begin{cases} 1 & \text{if } \exists s' \in L_\rho : s' \subseteq \zeta^*(c_1, \theta_1, \dots, c_l, \theta_l, s) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (10-19)$$

where ' \subseteq ' denotes a substring relationship.

The matching weight function is given in analogy to (10-16) by

$$\begin{aligned} \mu_{(c, \theta)^l}^2 : (C \times \Theta)^l \times \mathcal{P}(P) \times S &\rightarrow \mathbb{R}^+ \\ \mu_{(c, \theta)^l}^2(c_1, \theta_1, \dots, c_l, \theta_l, \rho, s) &=_{\text{def}} \begin{cases} \frac{\sum_{i=1}^l \mu_{c, \theta}(c_i, \theta_i, s)}{\sum_{i=1}^l v_{c, \theta}(c_i, \theta_i, s)} & \text{if } v_{(c, \theta)^l}^2(c_1, \theta_1, \dots, c_l, \theta_l, \rho, s) = 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (10-20)$$

An example shall clarify these definitions: Let

$$\rho_{JJ-NN} = \{\xi_1 \xi_2, \xi_2 \text{VB}[\text{RB}]^* \xi_1\}$$

be a set of syntactic regular expressions which defines the syntax of some legal adjective-noun phrases. The syntactic tag VB denotes a verb, RB denotes an adverb. The variables ξ_1 and ξ_2 mark the positions of legal matches for adjectives and nouns. The necessary tags and terms are given by the concept-tag pattern which is to be matched against a given sentence. The order of the concept-tag pairs in the pattern defines the meaning of ξ_1 and ξ_2 in the syntactic regular expressions. Let

$$cp = (\text{high}, \text{JJ}, \text{bandwidth}, \text{NN})$$

be such a concept-tag pattern. Variable ξ_1 will be set for a successful matching of the concept 'high' and the tag 'JJ' which denotes an adjective, ξ_2 marks a successful matching of the concept 'bandwidth' and the tag 'NN' which denotes a noun. According to the regular expressions in ρ_{JJ-NN} , a legal adjective-noun phrase may consist simply of an adjective directly followed by a noun or a noun followed by a verb, as many adverbs as desired, and finally by the adjective. Consider the sentence

$$s = (\text{The}, \text{DT}, \text{bandwidth}, \text{NN}, \text{is}, \text{VB}, \text{very}, \text{RB}, \text{high}, \text{JJ}).$$

In addition to the tags already known, 'DT' denotes a determiner. The substitution function $\zeta^*(cp, s)$ produces the string

$$\text{DT } \xi_2 \text{ VB RB } \xi_1$$

where ξ_2 indicates a successful matching of the concept ‘bandwidth’ accompanied by tag ‘NN’ and ξ_1 marks the position where (high, JJ) matches the given sentence s . Obviously, the second syntactic regular expression in the given set ρ_{JJ-NN} can produce the pattern

$$s' = \xi_2 \text{ VB RB } \xi_1$$

which is a substring of the substitution function’s result. Thus, $v_{(c,\theta)'}^2(cp, \rho_{JJ-NN}, s) = 1$.

10.6.2.4 Matching of Attribute-Value Pairs

The correct recognition and identification of attribute-value pairs in a sentence brings about the same problems as the identification of tagged concept patterns: It is first necessary to check whether an adjective found in a sentence refers to the noun that describes the given concept⁸. Thus, the first part of this matching problem is similar to the matching of tagged concept patterns as described above.

Recall that for a given attribute $a = (c, G[c], V)$ an *attribute-value pair* is a tuple (c, v) where $v \in V$. The set V contains all linguistic terms defined for a . Each linguistic term $v \in V$ is implicitly associated with a concept c_v where $c_v(v) = 1$ and $c_v(t) = 0$ for all terms $t \neq v$. In this work an attribute-value pair is an adjective-noun or numeral-noun phrase.

Let θ_{NN} and θ_{JJ} denote a noun tag and an adjective tag, respectively, and ρ_{AVP} be a set of syntactic regular expressions that produce legal adjective-noun phrases. For an attribute $a = (c, G[c], V)$ and a sentence s let $V'(a, s) \subseteq V$ be the set of linguistic terms which can be found in s and which grammatically refer to the attribute’s base concept c , i.e.

$$V'(a, s) =_{def} \{v' \in V \mid v_{(c,\theta)'}^2(c_{v'}, \theta_{JJ}, c, \theta_{NN}, \rho_{AVP}, s) = 1\}. \quad (10-21)$$

Note that numeral-noun phrases can be treated similarly. The matching frequency function for attribute-value pairs is defined by

$$v_{c,v}: C \times S \rightarrow \{0,1\}$$

$$v_{c,v}(c, s) =_{def} \begin{cases} 1 & \text{if } \exists a = (c, G[c], V) \wedge |V'(a, s)| > 0 \\ 0 & \text{otherwise} \end{cases} \quad (10-22)$$

The function $v_{c,v}$ is independent of the desired value v . This means that an attribute-value pair matches a sentence as soon as the sentence contains a noun term that matches the concept as well as an adjective that (a) syntactically refers to the concept of the attribute-value pair and (b) is the name of a defined linguistic term. This may be surprising but it ensures that non-exact matches of attribute values are found, too. The matching weight functions determines how well the found attribute-values match the desired value.

Let Ψ denote the set of all linguistic terms. Then, the matching weight function is defined as

$$\mu_{c,v}: C \times \Psi \times S \rightarrow [0,1]$$

$$\mu_{c,v}(c, v, s) =_{def} \max \{sub(F_{v'}, F_v) \mid v' \in V'(a, s) \wedge a = (c, G[c], V)\} \quad (10-23)$$

where $F_{v'}$ and F_v are the interpretations of v' and v and sub is the matching relation for fuzzy sets as defined by equation (10-8).

An example shall clarify these ideas. Consider the sentence

⁸ Note that a more elaborated extraction of attribute-value pairs is possible but not in the scope of this study.

Medium bandwidth applications are of a lower priority than high bandwidth applications.

Assume that an attribute for the base concept ‘bandwidth’ has been defined with

$$a = (\text{‘bandwidth’}, G[\text{‘bandwidth’}], \{\text{low}, \text{medium}, \text{high}\}).$$

and

$$(\text{‘bandwidth’}, \text{low})$$

is the attribute-value pair to be matched against the sentence. Given a suitable set of syntactic regular expressions (cf. section 10.6.2.3), the attribute-value pair matches, and $V(a,s) = \{\text{high}, \text{medium}\}$. Assume that the fuzzy sets in figure 10-7 (page 216) are the interpretations of the linguistic terms in V . Then,

$$\text{sub}(F_{\text{high}}, F_{\text{low}}) = 0 \text{ and } 0 < \text{sub}(F_{\text{medium}}, F_{\text{low}}) < 1.$$

The matching weight function returns the best match which in this case indicates a positive, yet not exact matching.

10.6.3 Matching Rules Against Documents

This section formally introduces the rule types that have been discussed in section 10.4.2. Rules actually define the attraction of documents. Recall that the document representatives in the semantic document space are moved according to the attractor approach presented in section 10.3. The algorithm for document movement assumes that a so-called ‘attractor-strength matrix’ is given that defines an ‘attractive force’ between each pair of documents. An attractor consists of two components: one component determines the attraction of the first document of the pair by the second, the other component determines the reverse attraction. Accordingly, each rule yields two matching components: In general rules consist of two sides, each of which is associated with one document of the pair considered. Syntactically, each rule side is a list of modeling elements along with weighting information which can be used to adjust the degree to which the document that is matched against this side attracts the document that is matched against the other side, provided that the rule triggers. Thus, rule triggering is asymmetric with respects to the given document pair in general. Each rule will define a fraction of the overall document attraction (which is the result of all rules that a pair of documents triggers).

In the following, four rule types ($\mathcal{R}_1 - \mathcal{R}_4$) along with the respective rule semantics will be introduced. Rule type \mathcal{R}_1 is the simple weighting of concepts, tagged concepts and attribute-value pairs as motivated in section 10.4.2.1. It will formally be defined as a special case of the more general rule type \mathcal{R}_2 (patterns of basic modeling elements, section 10.4.2.2) which is the first to be defined in the following. Rule type \mathcal{R}_3 realizes the sentence-oriented syntactic concept patterns from section 10.4.2.3 in its simple form, i.e. without additional grammatical constraints, whereas rule type \mathcal{R}_4 takes some additional constraints into account.

Formally, all rules will be interpreted by a trigger function $\tau: \mathcal{R} \times D \times D \rightarrow \mathbb{R}^+ \times \mathbb{R}^+$ which takes a rule $r \in \mathcal{R}$ and two documents d_1 and $d_2 \in D$ and yields a pair of positive real numbers that will be used later to set up the attractor-strength matrix. Note that due to the nature of rules for $\tau(r, d_1, d_2) = (a_{12}, a_{21})$ and $\tau(r, d_2, d_1) = (a'_{21}, a'_{12})$ in general $a_{12} \neq a'_{12}$ and $a_{21} \neq a'_{21}$.

10.6.3.1 Patterns of Basic Modeling Elements

Patterns of basic modeling elements form a document-oriented rule type. In general, they allow to express asymmetric relationships between documents. As motivated in section 10.4.2.2

the aim is to express that the occurrence of certain concepts or attribute-value pairs in one document and the occurrence of other concepts or attribute-value pairs in a second document causes an attraction between the respective document representatives in the semantic document space.

Let $\Gamma =_{\text{def}} C \cup (C \times V) \cup (C \times \Theta)$ denote the set of the two basic modeling elements, i.e. concepts and attribute-value pairs, as well as tagged concepts. To simplify the notation, for single concepts, tagged concepts and attribute-value pairs the matching frequency functions from sections 10.6.2.1, 10.6.2.2 and 10.6.2.4 can be summarized as follows:

$$\begin{aligned} v: \Gamma \times S &\rightarrow \mathbb{N}_0 \\ v(\gamma, s) &=_{\text{def}} \begin{cases} v_c(\gamma, s) & \text{if } \gamma \in C \\ v_{c,\theta}(\gamma, s) & \text{if } \gamma \in C \times \Theta \\ v_{c,v}(c, s) & \text{if } \gamma = (c, v) \in C \times V \end{cases} \end{aligned} \quad (10-24)$$

Similarly, the matching weight functions can be summarized by

$$\begin{aligned} \mu: \Gamma \times S &\rightarrow \mathbb{R}^+ \\ \mu(\gamma, s) &=_{\text{def}} \begin{cases} \mu_c(\gamma, s) & \text{if } \gamma \in C \\ \mu_{c,\theta}(\gamma, s) & \text{if } \gamma \in C \times \Theta \\ \mu_{c,v}(\gamma, s) & \text{if } \gamma \in C \times V \end{cases} \end{aligned} \quad (10-25)$$

Syntactically, a rule of this type is a tuple $r \in \mathcal{R}_2$ where

$$\mathcal{R}_2 =_{\text{def}} \Gamma^n \times [0, 1] \times \Gamma^m \times [0, 1]. \quad (10-26)$$

Given a rule $r = ((\gamma_{11}, \dots, \gamma_{1n}), \omega_1, (\gamma_{21}, \dots, \gamma_{2m}), \omega_2) \in \mathcal{R}_2$, for two given documents d_1 and d_2 the rule triggers if document d_1 matches all basic modeling elements $\gamma_{11}, \dots, \gamma_{1n}$ and document d_2 matches all basic modeling elements $\gamma_{21}, \dots, \gamma_{2m}$. The pattern $\gamma_{11}, \dots, \gamma_{1n}$ is referred to as the left-hand side of the rule which is matched against document d_1 . Accordingly, $\gamma_{21}, \dots, \gamma_{2m}$ is referred to as the right-hand side of the rule which is matched against document d_2 . The relative strength of triggering for each rule side can be controlled by the user-defined weights ω_1 and ω_2 . This adjustment allows to define whether the resulting attraction between d_1 and d_2 is balanced or unbalanced, i.e. whether the documents attract each other with the same strength (cf. section 10.3.3).

Formally, a rule of type \mathcal{R}_2 is interpreted by a trigger function

$$\begin{aligned} \tau_2: \mathcal{R}_2 \times D \times D &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \\ \tau_2(r, d_1, d_2) &\mapsto (a_{12}, a_{21}) \end{aligned} \quad (10-27)$$

where a_{12} and a_{21} are computed as follows: First, the document matching frequency of the respective basic modeling elements for each of the two given documents is determined, i.e.

$$\begin{aligned} f_{1j} &=_{\text{def}} \sum_{s \in d_1} v(\gamma_{1j}, s) \text{ for } j = 1, \dots, n, \\ f_{2k} &=_{\text{def}} \sum_{s \in d_2} v(\gamma_{2k}, s) \text{ for } k = 1, \dots, m. \end{aligned} \quad (10-28)$$

Thus, the value f_{1j} denotes the matching frequency of the j -th modeling element γ_{1j} from the left-hand side of the rule which is related to d_1 (f_{2j} accordingly for the right-hand side). Ac-

according to the informal presentation of this rule type in section 10.4.2.2, for both sides of the rule each modeling element in the corresponding pattern has to match the respective document d_i . Consequently, in case that a certain modeling element of either side of the rule does not match its document, there will be no attraction between d_1 and d_2 (or in other words: the rule does not match the given pair of documents). Formally,

$$a_{12} = a_{21} =_{def} 0 \quad \text{if } \min(f_{1j}, f_{2k})_{j=1..n, k=1..m} = 0. \quad (10-29)$$

Otherwise, all frequencies are greater zero and the average matching weight of all modeling elements for the respective document is defined by

$$\begin{aligned} m_{1j} &=_{def} \frac{1}{f_{1j}} \cdot \sum_{s \in d_1} \mu(\gamma_{1j}, s) \quad \text{for } j = 1, \dots, n, \\ m_{2k} &=_{def} \frac{1}{f_{2k}} \cdot \sum_{s \in d_2} \mu(\gamma_{2k}, s) \quad \text{for } k = 1, \dots, m. \end{aligned} \quad (10-30)$$

In order to eliminate the influence that the document length might have on the matching frequency, the calculated values f_i are divided by the number of sentences in the respective document, i.e.

$$\begin{aligned} \hat{f}_{1j} &=_{def} \frac{f_{1j}}{|d_1|} \quad \text{for } j = 1, \dots, n, \\ \hat{f}_{2k} &=_{def} \frac{f_{2k}}{|d_2|} \quad \text{for } k = 1, \dots, m. \end{aligned} \quad (10-31)$$

In this work, frequency normalization uses the number of sentences instead of the number of words in the documents because a sentence is regarded to be a more significant structure than a single word. As this rule type is matched against words, however, the normalized frequency could theoretically be greater than 1 – an effect which will be adjusted later.

Finally, the resulting matching frequency of each rule side is defined as the average of all normalized matching frequencies of each modeling element from the respective side of the rule, i.e.

$$\begin{aligned} \hat{f}_1 &=_{def} \frac{1}{n} \cdot \sum_{j=1}^n \hat{f}_{1j}, \\ \hat{f}_2 &=_{def} \frac{1}{m} \cdot \sum_{k=1}^m \hat{f}_{2k}. \end{aligned} \quad (10-32)$$

Similarly, the average matching weights of the single modeling elements are averaged for each side of the rule, i.e.

$$\begin{aligned} m_1 &=_{def} \frac{1}{n} \cdot \sum_{j=1}^n m_{1j}, \\ m_2 &=_{def} \frac{1}{m} \cdot \sum_{k=1}^m m_{2k}. \end{aligned} \quad (10-33)$$

Finally, the pair $\tau_2(r, d_1, d_2) = (a_{12}, a_{21})$ is defined by

$$a_{12} =_{\text{def}} \omega_1 \cdot m_1 \cdot m_2 \cdot \frac{\hat{f}_1 + \hat{f}_2}{2},$$

$$a_{21} =_{\text{def}} \omega_2 \cdot m_1 \cdot m_2 \cdot \frac{\hat{f}_1 + \hat{f}_2}{2}.$$
(10-34)

Figure 10-9 depicts the effect of applying a simple rule of this type to two documents and illustrates that the trigger function τ_2 is not symmetric regarding its parameters d_1 and d_2 .

According to the informal presentation of proposed rule types a simple and useful means for defining a bias towards certain key topics in documents is to simply ‘stress’ concepts or attribute-value pairs (cf. section 10.4.2.1). Since this is supposed to be an important rule class in practice it is reasonable to define a rule type

$$\mathcal{R}_1 =_{\text{def}} \Gamma \times [0,1] \quad (10-35)$$

as a special case of rule type \mathcal{R}_2 . A rule $r = (\gamma, \omega) \in \mathcal{R}_1$ is interpreted by a trigger function

$$\tau_1: \mathcal{R}_1 \times D \times D \rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \quad (10-36)$$

$$\tau_1(r, d_1, d_2) = \tau_1((\gamma, \omega), d_1, d_2) =_{\text{def}} \tau_2((\gamma, \omega, \gamma, \omega), d_1, d_2).$$

According to equation (10-34), $\tau_1(r, d_1, d_2)$ yields a pair (a_{12}, a_{21}) where $a_{12} = a_{21}$ since $\omega = \omega_1 = \omega_2$. In other words, the simple weighting of concepts, tagged concepts or attribute-value pairs always results in a balanced attraction between matching pairs of documents.

10.6.3.2 Simple Syntactic Concept Patterns

Rule types \mathcal{R}_1 and \mathcal{R}_2 are document-oriented, i.e. the respective trigger functions check whether a given pair of documents contains the modeling elements declared in the considered rule. The modeling elements do not have to appear all in one sentence. In contrast, the remaining rule types are sentence-oriented, i.e. a given sequence of concepts or tagged concepts matches a document only if the complete sequence appears within a sentence.

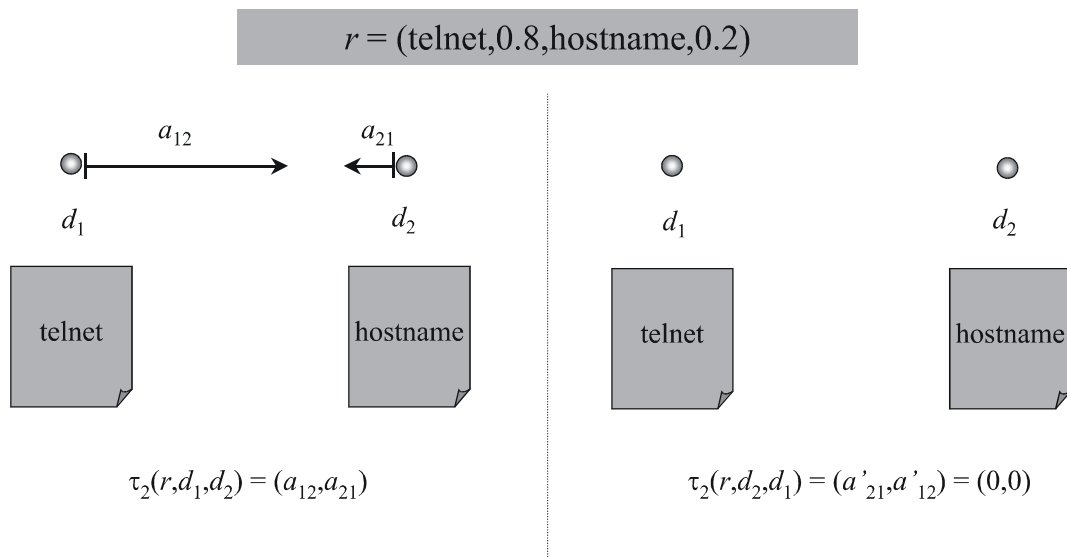


Figure 10-9: Triggering of rule $r = (\text{'telnet'}, 0.8, \text{'hostname'}, 0.2) \in \mathcal{R}_2$ with $m=n=1$ and $\gamma_{11}, \gamma_{21} \in C$ for a document containing the concept ‘telnet’ (d_1) and a document containing the concept ‘hostname’ (d_2). On the left, the left-hand side of the rule is applied to d_1 and the right-hand side to d_2 , resulting in values $0 < a_{12} = 4 \cdot a_{21}$. On the right, the rule is matched against the documents in reverse order. Here, neither side of the rule triggers since d_1 (d_2) does not contain the concept ‘hostname’ (‘telnet’).

A syntactic concept pattern is an asymmetric rule type which allows to express a relationship between sequences of concepts with syntactic information (cf. section 10.4.2.3). As motivated in section 10.2.2, a simple form of syntactic pattern matching is introduced first.

The signature of rules of this type is given by

$$\mathcal{R}_3 =_{\text{def}} (C \times \Theta)^n \times [0,1] \times (C \times \Theta)^m \times [0,1] \quad (10-37)$$

where Θ is the set of all part-of-speech tags. Thus, the rule contains two lists of concept-tag pairs and two weights. The first list represents the left-hand side of the rule which is matched against the first given document, the second list is the right-hand side which is matched against the second given document (similar to patterns of basic modeling elements). Again, the user may define weights to adjust the degree of triggering for both sides of the rule. Assume that Θ contains a special joker tag which matches all other syntactic tags (this allows an even simpler matching without tag information).

A rule of type \mathcal{R}_3 is interpreted by a trigger function

$$\begin{aligned} \tau_3: \mathcal{R}_3 \times D \times D &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \\ \tau_3(r, d_1, d_2) &\mapsto (a_{12}, a_{21}) \end{aligned} \quad (10-38)$$

where, similar to section 10.6.3.1, a_{12} and a_{21} are computed based on the document matching frequency and the average matching weight of the respective tagged concept pattern. For this, the matching frequency function $v_{(c,\theta)^i}^1$ and the matching weight function $\mu_{(c,\theta)^i}^1$ from section 10.6.2.3 are used.

Given a rule $r = (cp_1, \omega_1, cp_2, \omega_2) \in \mathcal{R}_3$ with $cp_1 =_{\text{def}} (c_{11}, \theta_{11}, \dots, c_{1n}, \theta_{1n})$, $cp_2 =_{\text{def}} (c_{21}, \theta_{21}, \dots, c_{2m}, \theta_{2m})$ and two documents $d_1, d_2 \in D$, the matching frequency f_i and the average matching weight m_i for document d_i , $i = 1..2$, is defined by

$$f_i =_{\text{def}} \sum_{s \in d_i} v_{(c,\theta)^i}^1(cp_i, s) \quad (10-39)$$

and

$$m_i =_{\text{def}} \begin{cases} \frac{1}{f_i} \cdot \sum_{s \in d_i} \mu_{(c,\theta)^i}^1(cp_i, s) & \text{if } f_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (10-40)$$

The normalized matching frequency \hat{f}_i is computed according to

$$\hat{f}_i =_{\text{def}} \frac{f_i}{|d_i|}, i = 1..2. \quad (10-41)$$

Note that $\hat{f}_i \in [0,1]$ since a tagged concept pattern may only match once per sentence at most. Finally, the result $\tau_3(r, d_1, d_2) = (a_{12}, a_{21})$ of the trigger function is defined according to equation (10-34).

As an example consider the rules

$$r_1 = ((\text{send}, \text{VB}, \text{message}, \text{NN}), 1.0, (\text{receive}, \text{VB}, \text{message}, \text{NN}), 1.0)$$

and

$$r_2 = ((\text{send}, *, \text{message}, *), 1.0, (\text{receive}, *, \text{message}, *), 1.0)$$

where ‘*’ denotes the joker tag.

The trigger function $\tau_3(r_1, d_1, d_2)$ defines an attraction between documents d_1 and d_2 only if d_1 contains at least one sentence where ‘send’ appears as a verb (denoted by tag ‘VB’) and ‘message’ as a noun (denoted by ‘NN’) and d_2 contains at least one sentence where ‘receive’ appears as a verb and ‘message’ as a noun⁹. In all other cases, $\tau_3(r_1, d_1, d_2) = (0, 0)$. Note that it may be the case that the verbs and nouns are grammatically not related to each other (cf. section 10.6.2.3). Rule r_2 triggers already if d_1 contains the concepts ‘send’ and ‘message’ in a common sentence and d_2 has a sentence where ‘receive’ and ‘message’ appear together, regardless of the actual grammatical category.

10.6.3.3 Syntactic Concept Patterns with Constraints

This rule type is very similar to simple syntactic concept patterns. However, additional syntactical constraints for the matching of tagged concepts are given. Whereas the rule type from section 10.6.3.2 uses the simple version of the matching frequency function and the matching weight function, respectively, in this case constraints in form of syntactic regular expressions are respected by the matching functions.

Consequently, the rule type from above is extended so that a syntactic concept pattern with constraints is a tuple $r \in \mathcal{R}_4$ with

$$\mathcal{R}_4 =_{\text{def}} (C \times \Theta)^n \times [0, 1] \times (C \times \Theta)^m \times [0, 1] \times \mathcal{P}(\mathcal{P}). \quad (10-42)$$

where $\mathcal{P}(\mathcal{P})$ denotes the power set of syntactic regular expressions. A rule $r \in \mathcal{R}_4$ is interpreted by the trigger function

$$\begin{aligned} \tau_4: \mathcal{R}_4 \times D \times D &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \\ \tau_4(r, d_1, d_2) &\mapsto (a_{12}, a_{21}). \end{aligned} \quad (10-43)$$

Compared to rules of type \mathcal{R}_3 only the definition of the matching frequency and the matching weight changes. For a given rule $r = (cp_1, \omega_1, cp_2, \omega_2, \rho)$, where $\rho \subseteq \mathcal{P}$, f_i and m_i , $i = 1, 2$, are computed according to

$$f_i =_{\text{def}} \sum_{s \in d_i} v_{(c, \theta)^i}^2(cp_i, \rho, s) \quad (10-44)$$

and

$$m_i =_{\text{def}} \begin{cases} \frac{1}{f_i} \cdot \sum_{s \in d_i} \mu_{(c, \theta)^i}^2(cp_i, \rho, s) & \text{if } f_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (10-45)$$

Finally, the trigger function’s result $\tau_4(r, d_1, d_2) = (a_{12}, a_{21})$ is computed as in section 10.6.3.2.

10.6.4 Defining the Attractor Strength Matrix on Basis of the Rules

Recall that an attractor $\phi_{i \leftrightarrow j}$ between document representatives \mathbf{x}_i and \mathbf{x}_j from the semantic document space X is modeled by a tuple of real numbers $\phi_{i \leftrightarrow j} = (\phi_{ij}, \phi_{ji}) \in [0, 1]^2$ where ϕ_{ij} describes the strength with which \mathbf{x}_i is attracted by \mathbf{x}_j and ϕ_{ji} describes the strength with which \mathbf{x}_j is attracted by \mathbf{x}_i .

⁹ Recall that terms in concepts and documents are compared regarding their word stem.

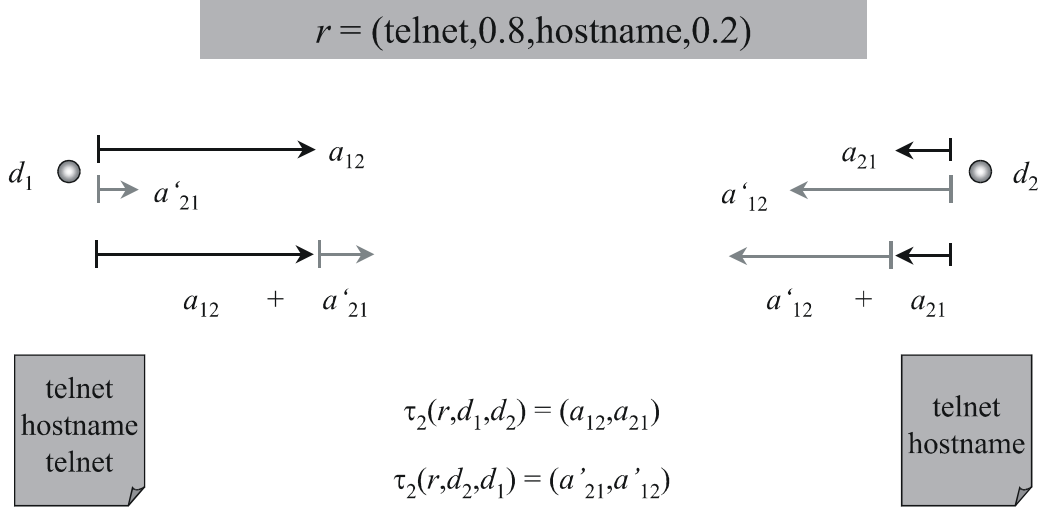


Figure 10-10: Summing up the results of rule triggering for a given pair of documents. The ‘telnet’ side of the rule triggers stronger for d_1 than for d_2 .

Concerning the rule approach, so far there is a trigger function for each rule type which determines the strength to which each document of a given pair attracts the other regarding to a single rule. Now it is necessary to cumulate these fractions of attraction and to construct the attractor-strength matrix. Since rule type \mathcal{R}_1 is just a special case of \mathcal{R}_2 , the following definitions can be limited to \mathcal{R}_2 , \mathcal{R}_3 , and \mathcal{R}_4 .

Let $\mathcal{R} =_{\text{def}} \mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$. Let τ summarize the single trigger functions, i.e.

$$\begin{aligned} \tau: \mathcal{R} \times D \times D &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \\ \tau(r, d_1, d_2) &=_{\text{def}} \tau_i(r, d_1, d_2) \text{ for } r \in \mathcal{R}_i, i = 2, 3, 4 \end{aligned} \quad (10-46)$$

The trigger function $\tau(r, d_1, d_2)$ yields a pair (a_{12}, a_{21}) where (intuitively) a_{12} describes the strength with which d_2 attracts d_1 (a_{21} accordingly) regarding rule r . Recall that the result of τ depends on the order of the documents which are considered for a rule r . In particular, for $\tau(r, d_1, d_2) = (a_{12}, a_{21})$ and $\tau(r, d_2, d_1) = (a'_{21}, a'_{12})$ in general $a_{12} \neq a'_{12}$ and $a_{21} \neq a'_{21}$ (cf. figure 10-9). In other words, each rule can be applied twice for a given pair of documents since its semantics depends on the order of documents considered. Thus, it is necessary to add the results of both rule applications. The resulting attraction between a pair of documents regarding rule r is defined as

$$\begin{aligned} \alpha: \mathcal{R} \times D \times D &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \\ \alpha(r, d_1, d_2) &=_{\text{def}} \tau(r, d_1, d_2) \oplus \tau(r, d_2, d_1) \end{aligned} \quad (10-47)$$

where $(a_{12}, a_{21}) \oplus (a'_{21}, a'_{12}) =_{\text{def}} (a_{12} + a'_{12}, a_{21} + a'_{21})$. Obviously, it holds for $\alpha(r, d_1, d_2) = (a_{12}, a_{21})$ and $\alpha(r, d_2, d_1) = (a'_{21}, a'_{12})$ that $a_{12} = a'_{12}$ and $a_{21} = a'_{21}$. Figure 10-10 depicts the asymmetric results of a trigger function and the resulting document attraction regarding a single rule.

Now the resulting attractor for each pair of documents has to be determined which takes into calculation all rules defined in a given rule base. This is done in two steps: First, a matrix of cumulated rule results is defined. Second, this matrix is normalized so that a legal attractor-strength matrix is gained.

Let D denote the considered collection of documents and $R \subseteq \mathcal{R}$ be a rule base. An element $d \in D$ is a formal document representation (according to section 10.6.2). Let $N(r)$ denote the

number of documents from D that trigger rule $r \in R$. Then, a matrix $\Phi' = (\phi'_{ij})$, $1 \leq i, j \leq |D|$ can be defined where

$$(\phi'_{ij}, \phi'_{ji}) =_{\text{def}} \sum_{r \in R} \alpha(r, d_i, d_j) \cdot \log\left(\frac{|D|}{N(r)}\right) \text{ for } d_i, d_j \in D, i \neq j \quad (10-48)$$

and $\phi'_{ij} =_{\text{def}} 0$ for $i = j$.

The matrix Φ' contains the cumulated result of applying all rules from R to all documents from D . The logarithmic factor included in the formula weakens the effect of rules that trigger for most documents in the collection and strengthens the effect of rules that trigger in only a few documents. This effect is generally desired: Since rules result in document attraction, a rule that triggers for all documents would cause all documents to move towards each other more or less fast. This would provide only limited additional structure information and is generally an undesired effect.

Since the sums ϕ'_{ij} and ϕ'_{ji} have no upper limit it is necessary to normalize these values. So finally, the attractor-strength matrix $\Phi = (\phi_{ij})$, $1 \leq i, j \leq |D|$, $\phi_{ij} \in [0, 1]$ contains the elements

$$\phi_{ij} =_{\text{def}} \frac{\phi'_{ij}}{\max(\phi'_{ij})_{i,j=1..|D|}}. \quad (10-49)$$

The construction of Φ finishes the process of analyzing documents from the collection. The defined attractive forces will move the documents in the semantic document space according to algorithm 10-1.

10.7 Complexity of the Semantic Refinement Module

Algorithm 10-2 summarizes the process of calculating the attractor-strength matrix Φ based on a given document collection D and a rule base R . Before Φ is computed, each document $d_i \in D$ is matched against both sides of each rule r_k , and the respective matching frequency \hat{f} and matching weight m as defined by the different trigger functions is computed. To store this data a $|D| \times |R|$ -matrix M is necessary where $M(i, k)$ stores the matching frequency and the matching weight of document d_i for both, the left and the right hand-side of rule r_k . After this preparation, M contains all information necessary for calculating the attractor-strength matrix

Algorithm 10-2: Computing the attractor-strength matrix for a document collection

```

CalculateAttractors (D,R)
BEGIN
  FOR each rule  $r_k \in R$  DO
    FOR each document  $d_i \in D$  DO
      BEGIN
        match both sides of  $r_k$  against  $d_i$  and store the
        respective matching frequency and matching weight
        in  $M(i, k)$ ;
      END
    calculate  $\Phi$  according to equation (10-49);
  return  $\Phi$ ;
END

```

Φ according to equation (10-49).

Now, the complexity of the approach can be discussed. Matching concepts and attribute-value pairs against sentences (as defined in section 10.6.2) requires linear time regarding the sentences' length $|s|$. More detailed, matching of concepts (10.6.2.1) and tagged concepts (10.6.2.2) can be realized in time $O(|s| \cdot |c|)$ where $|c|$ denotes the number of terms in concept c , i.e. $|c| = |\{t|c(t)>0\}|$. Usually, a concept contains a small number of terms which can be regarded as constant. Matching tagged concept patterns and attribute-value pairs (sections 10.6.2.3 and 10.6.2.4) requires the matching of regular expressions against sentences which can be done in time $O(|s|)$. In addition, calculating the matching weight of attribute-value pairs requires a matching of trapezoid fuzzy set according to equation (10-8) which requires constant time. For practical document collections the length of sentences can be regarded as constant.

The rule trigger functions basically require the matching of their modeling elements against all sentences of a document. This can be done in time $O(\tilde{s})$ where \tilde{s} denotes the average number of sentences in a document. According to algorithm 10-2 the preparatory step of rule evaluation can be realized in time $O(|R| \cdot |D| \cdot \tilde{s})$. The remaining calculation of the attractor-strength matrix Φ according to equation (10-49) obviously requires time $O(|D|^2)$.

Turning back to the document movement in the semantic document space (algorithm 10-1 in section 10.3.2), the execution time depends on the number of iteration steps, t_{\max} , and the number of documents. Obviously, the algorithm requires $O(t_{\max} \cdot |D|)$ calculations of equation (10-4) which can be solved in time $O(|D|)$, i.e. $O(t_{\max} \cdot |D|^2)$ steps in total.

10.8 Reflection of the General Design

This section sums up and reflects the general approach of the semantic refinement module. The basic idea is to incorporate adaptability by a rule-based approach: In addition to the known document relationships a personal 'degree of document relatedness' can be defined via rules. These rules define 'attractive forces' between documents which are interpreted by moving the document representatives in the semantic document space. Thus, a bias is formed by 'superimposing' a view which is expressed in rules. The concept of 'attractive forces' between document representatives brings along an intuitive metaphor for structure modification. Besides, the chosen extension of the architecture of the basic framework maintains the modularity of the proposed document map approach.

In section 10.2.2 some requirements for the rule approach have been defined. First, a simple, intuitive rule language with expressive modeling elements and formal semantics of rule types is demanded. Since each rule type directly contributes to the attraction of document pairs the effect of a rule can be intuitively understood by considering the movement metaphor. Rule types which are interesting for managing the knowledge contained in specialized document collections have been motivated in section 10.4 and realized afterwards. The approach is more expressive than simple term matching: Terms can be flexibly combined and weighted to domain-specific concepts, and attribute-value pairs can be defined which allow to integrate numerical information. Moreover, syntactic information can be used to restrict possible term matchings. The semantics of the rules and the matching is clearly defined in section 10.6.

In the semantic analysis component proposed here, the set of concepts does not induce a structured concept space (in the sense of a concept hierarchy or a semantic network). This is acceptable for the considered application domain: Since it is the intention to allow personal

views and bias-forming rules, it should be possible to define concepts subjectively, task-dependent and dynamically. Formal taxonomies would be a burden in such a highly flexible context. The concepts defined here merely form a simple fuzzy thesaurus without inter-concept relationships.

Finally, consider the demand for a staged approach, i.e. the possibility to define simple rules which can be enriched by more semantic and syntactic information if necessary: Some rule modeling elements and rule types are proposed in a simple as well as in a more elaborated form (cf. sections 10.6.2.3, 10.6.3.2, 10.6.3.3), thus enabling to enhance definitions if desired. Furthermore, the varying granularity of the different rule-types allows the definition of rules with growing complexity and expressiveness. Consider the following proposal for applying the rule approach in different, qualitatively increasing stages:

- **Stage 1:** Stressing of important concepts (using rule type \mathcal{R}_1). This stage can make use of an existing domain glossary to weight important concepts higher than others. It is also interesting for a first realization of a view by utilizing and re-using application- and task-specific weighting rules that have been defined for similar tasks.
- **Stage 2:** Definition of document-oriented patterns of basic modeling elements (using rule type \mathcal{R}_2). This allows to define rules that establish relationships between arbitrary concepts – an essential feature for expressing views. Furthermore, attribute-value pairs could be used to define more subtle relationships between texts.
- **Stage 3:** Definition of syntactic patterns (rule types \mathcal{R}_3 and \mathcal{R}_4) for a more detailed and precise specification of semantic relationships.

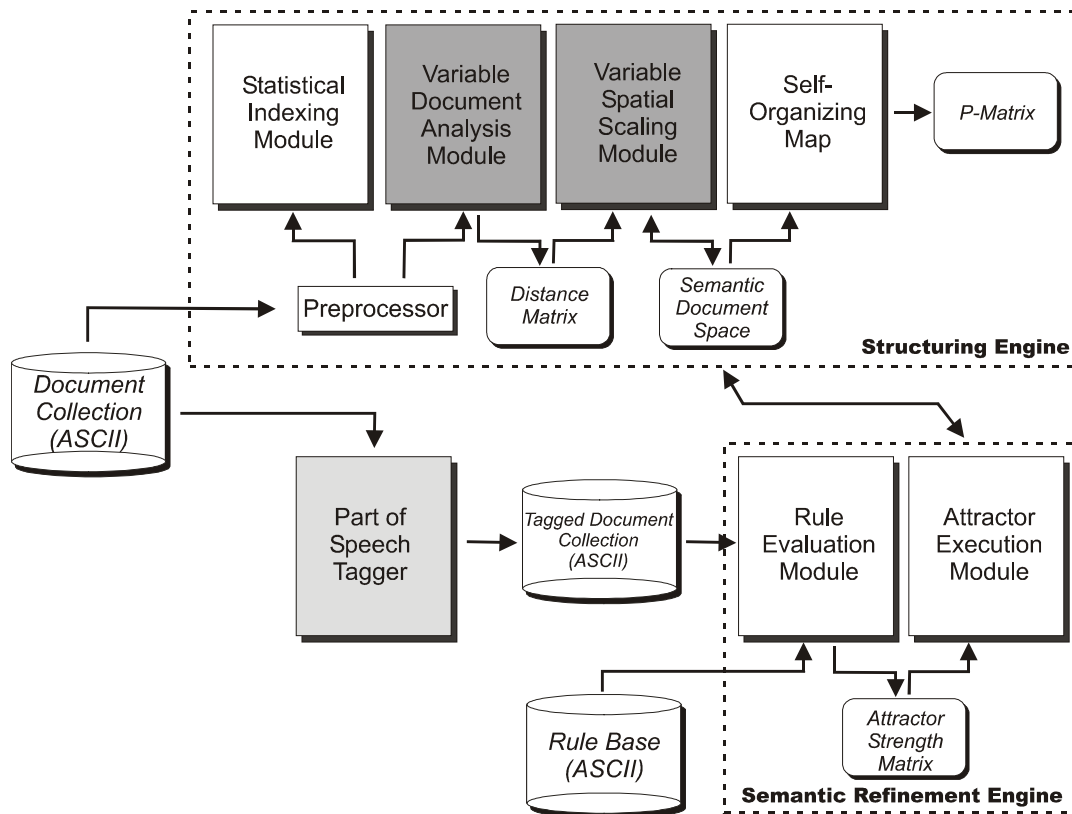


Figure 10-11: Extended architecture of the document map system DocMINER

10.9 Integration into the DocMINER System

This section presents the integration of the semantic refinement module into the DocMINER system and sketches additional tools which help to define rule bases and to understand the effects of the document movement algorithm.

10.9.1 Extended Architecture

Figure 10-11 presents the extended architecture of the system (cf. chapter 7.2). Analogous to the structuring engine which implements the basic framework, the semantic refinement engine which implements the extension presented in this chapter is realized as a dynamic link library. A part-of-speech tagger (an external tool) preprocesses the document collection by assigning syntactical tags to every word of each document. The tagged collection and a rule base (which respects the syntax defined in appendix F) are taken as input by the refinement engine's rule evaluation module which calculates the attractor strength matrix (cf. section 10.3.2). The attractor execution module performs the actual document movement by manipulating the semantic document space which is managed by the structuring engine (this data exchange is realized via a written data-file).

10.9.2 Extended Workspace and Additional Tools

Figure 10-12 shows the extended workspace of DocMINER. Existing document maps which have been generated by the realization of the basic framework can be adapted to the special

interest of the analyst by using the semantic refinement module. As additional input to be specified in a parameter form the module requires (a) a document collection which is syntactically tagged by a part-of-speech tagger and (b) a rule base which respects the syntax defined in appendix F. Additional parameters which have to be specified by the user include the number of iterations for the document movement and the movement length factor p , specifying the amount of movement to be performed (see section 10.3.2 for a more detailed explanation).

There are two additional tools which are of help for semantically refining document maps: The first is an editor which assists the user in the creation of a rule base (figure 10-13), using a tree view to display the current data objects. The user can edit objects, add objects to the tree or remove them. Fuzzy sets which represent vague attribute values are visualized to allow an intuitive and easy definition of the value sets and their overlapping properties. The second tool helps to understand the effect of varying the parameters of the document movement algorithm by simulating and visualizing object movement in two-dimensional space. The user can interactively place document representatives on a pad and edit an attractor strength matrix. After specifying the run-time parameters p and the number of iterations, each step of the iterated movement is visualized, showing the movement path of each object. Different scenarios can be stored and restored from disk. Figure 10-14 shows a screenshot of the tool.

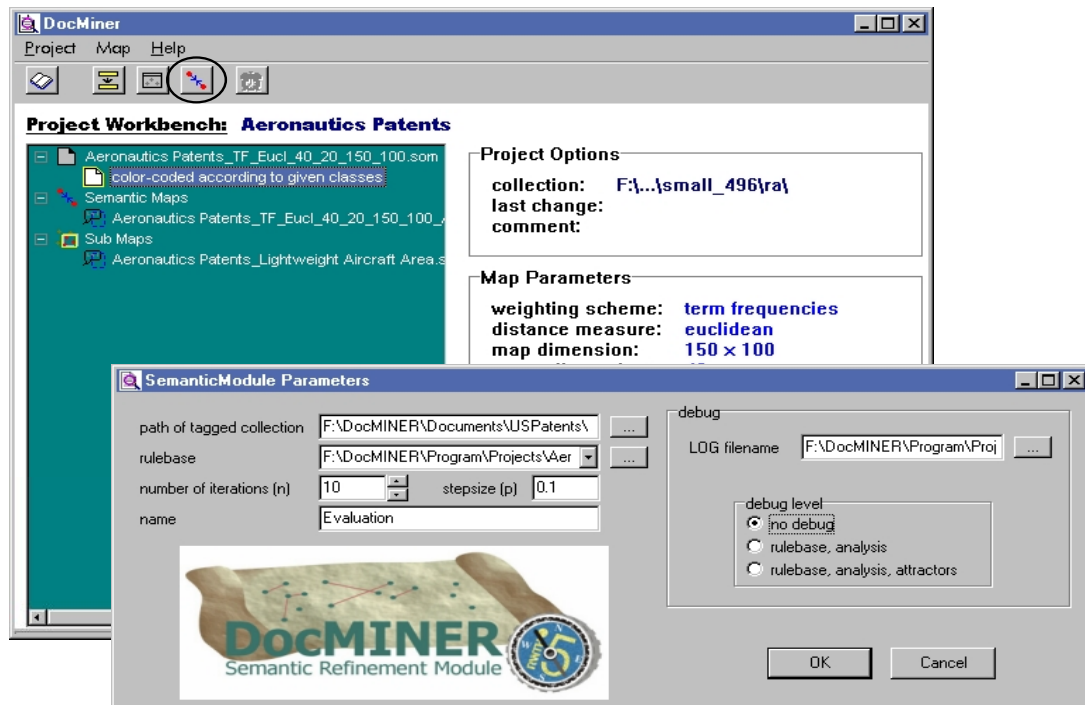


Figure 10-12: DocMINER's project workbench, definition form 'semantic module parameters'

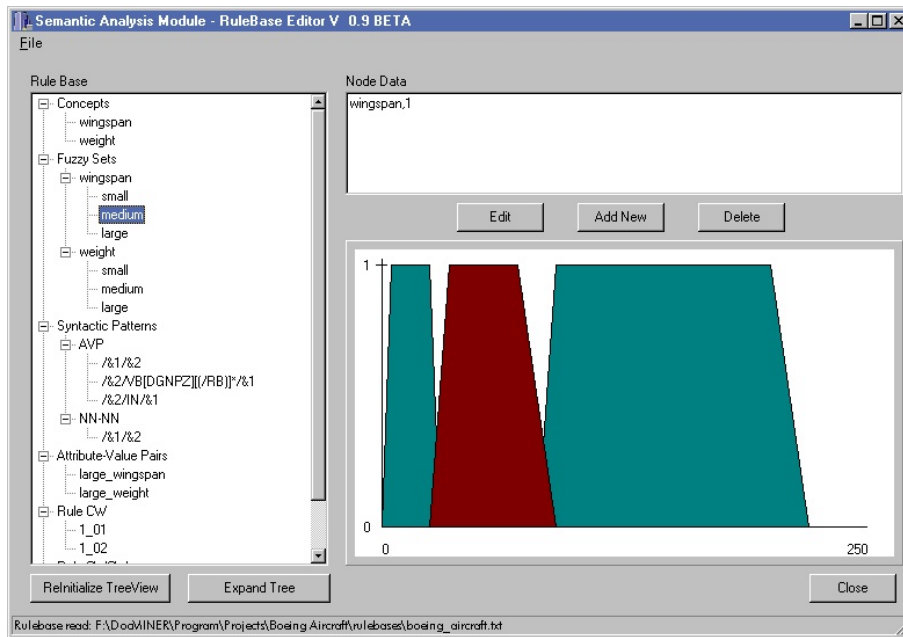


Figure 10-13: Rule base editor for the semantic refinement module. The left hand side presents a tree view of the rule base, the text field allows to edit items, and the graphic panel visualizes fuzzy attribute values (here: small, medium and large for the attribute 'wingspan').

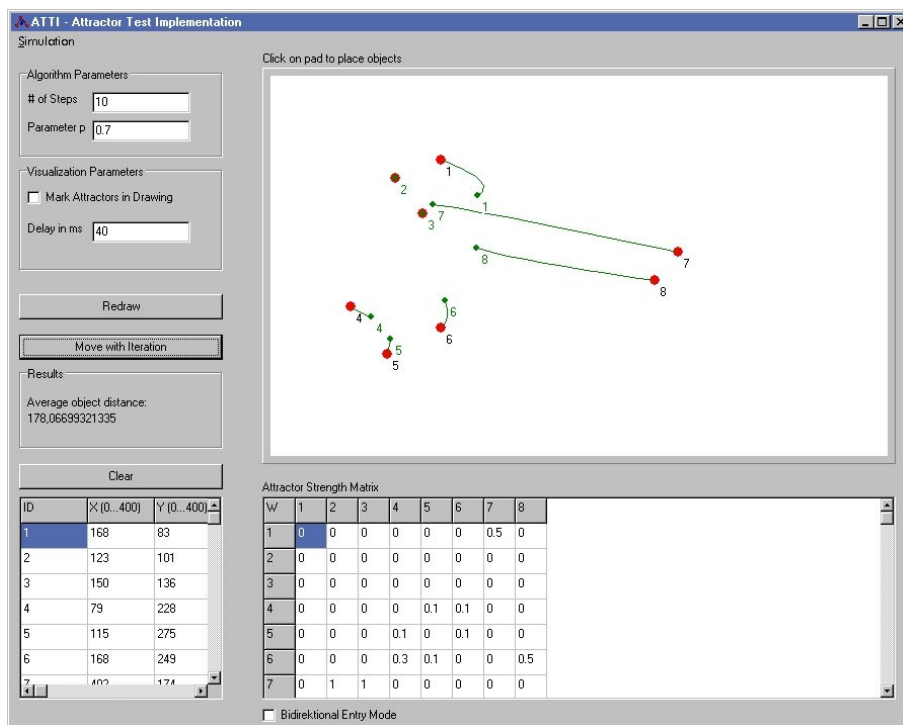


Figure 10-14: Interactive tool for visualizing object movement in two-dimensional space according to the document movement algorithm of the semantic refinement module

10.10 Experiments

This section studies whether structures provided by a basic document map can be meaningfully influenced by the semantic refinement method proposed in this chapter. In the first experiment the approach is applied to a test collection for which a detailed classification of

documents is available (similar to chapter 6.5). Rules are defined that aim at influencing the structure presented by a basic document map in a pre-defined way so that the achieved effects can be considered objectively. The focus of the second experiment is the evaluation of a special subset of rules, namely rules based on attribute-value pairs. The relatively small test collection is therefore analyzed manually in order to assess the results achieved by applying the semantic refinement approach.

10.10.1 Experiment I: Aeronautics Patent Abstracts

In this experiment the basic map approach is applied to a collection of aeronautics patents first. Furthermore, a ‘view’ on the map is created using the semantic refinement module. The text corpus consists of a set of U.S. patent abstracts publicly available from the IBM patent database server¹⁰. The U.S. patents are stored within a class hierarchy out of which a subset of *Class 244 – Aeronautics* has been taken. The subclasses which are included in the collection are shown in figure 10-15. The collection considered here consists of 496 documents (an additional experiment with 1,488 documents of this class can be found in appendix G).

Figure 10-16 (a) shows the original document map with no rules applied. Details on parameters for map calculation can be found in table 10-5 (p. 244). Each document on the map has been colorized according to the document class it belongs to. Note that some patents belong to multiple classes. In such cases the document color has been chosen arbitrarily. It is clearly recognizable that there are regions where certain classes have been grouped together while other classes are spread all over the map.

To refine the structures a rule base for the semantic refinement module has been created with the following objectives in mind:

- Improve the map by setting up additional semantic rules that help distinguish some of the different classes. This allows to objectively show “map enhancement” by the semantic refinement module since the resulting structures can be compared to the given class structure.
- Focus on special areas of interest. In this case documents dealing with helicopters and patents about missiles and rockets are weighted.

In order to achieve these results rules have been created that express relationships for the classes shown in table 10-1. In addition to these class modeling rules, a set of rules for expressing inter-class relationships (2_01, 3_09, 3_10) and a set for setting a focus on the special interest in helicopters and rocket driven technology (1_03, 1_04, 1_05, 2_02) have been defined. The rule base is presented in table 10-2.

Consider the refined map presented in figure 10-16(b). The structure of this map is much clearer than that of the pre-structured map. The document subclasses that have been stressed by the rules stick out and the map reflects the modeled inter-class relationships. Consider the lightweight aircraft class (class 900ff.). The corresponding documents lie significantly closer together and have moved towards the balloon and airship classes as desired. The aircraft control classes (075, 220ff, 87ff., 091, cf. figure 10-16(c) and figure 10-15), which have been spread across the whole map before, now form a visible cluster in the north-eastern area of the map. Of course there are still a lot of outliers within this class, but this is due to the topic of these documents; aircraft control is a topic that occurs within almost every aspect of aeronau-

¹⁰ www.patents.ibm.com as of April 2000

tics. Visibly, the evacuation slides group (north-eastern edge) has become smaller. The reason is that a lot of corresponding documents fall into one spot on the map since these texts have already been close to each other before rule application (cf. central area in figure 10-16(a)).

The map also clearly reflects the special interest in documents about helicopter and rocket driven technologies (cf. figure 10-17). Since there are no specific classes for these documents it is necessary to search for matching documents by appropriate queries. In the pre-structured map, documents about rockets and missiles are scattered across the map. In the refined map relevant documents can be found in areas that form well distinguishable spots. It is interesting to note that the rocket driven technology group lies close to the “Propulsion: Launching” set of documents, which obviously makes sense. This effect is due to the attraction of documents that contain the concept ‘launch’.

A closer analysis of the map’s microstructure before and after the semantic refinement has been performed in [Tusk00] in order to evaluate the consistency of the structural changes. It turned out that the structure indeed has been improved with respect to the view since it reflects both, the pre-structuring and the rule-based refinement. More specifically, the relation of documents that are nearby each other in the refined map but more distant in the pre-structured map and vice versa can be understood by considering triggering rules and the similarity measure used for pre-structuring.

006	Aircraft, heavier than air
007.R	Convertible
030	Aircraft, lighter than air, Airships
031	Aircraft, lighter than air, Balloons
036	Aircraft, lighter than air, Airship and helicopter sustained
053	Aircraft power plants
062	Aircraft propulsion
	(063) Launching, (064) Manual,
065	Screw
	(066) Tilting, (067) Body encircling, (068) Elongated,
	(069) Contra-propeller arrangements
	(070) Paddle Wheel, (071) Reciprocating propeller, (072) Beating wing
073	Fluid
073.B	Vacuum induced by radial flow
073.C	Radial outward and downward flow
074	Explosive Jet
075	Aircraft control
075.A	Flutter prevention
087	Rudders and empennage
088	Rudders universally mounted
089	Elevators both front and rear
090	Ailerons and other roll control devices
090.A	Balanced air pressure
090.B	Roll control spoilers
091	Vertical fins
092	Stabilizing propellers
093	Stabilizing weights
	(094) Ballast storage and release, (095) Ballast making
096	Airship control
	(097) Buoyancy varying, (098) Gas bag inflation, (099) Gas release
220	Pilot operated
221	Control system
	(223) With feel, (232) With cable and linkage,
	(228) Electric
234	Controller
235	Rudder bar and pedal
900	Lightweight, winged, air vehicle (ultralight or hang glider)
901	Having delta shaped wing
902	Having parachute type wing
903	Powered
904	Miscellaneous hardware or control
905	Inflatable Evacuation Slides

Figure 10-15: Class structure of aeronautics collection

Table 10-1: Classes for which rules are defined

class name	class numbers	rules
<i>aircraft, lighter than air, airships</i>	030	1_02
<i>aircraft, lighter than air, balloons</i>	031	1_01
<i>lightweight, winged, air vehicle (ultralight or hang glider)</i>	900, 901, 902, 903, 904	1_07, 1_08, 2_03, 3_07, 3_08
<i>inflatable evacuation slides</i>	905	3_01, 3_02, 3_03, 3_04, 3_05
<i>aircraft control - vertical fins</i>	091	3_12, 3_13, 3_14, 3_15
<i>aircraft control - general</i>	075ff, 220ff	3_16
<i>aircraft propulsion - general</i>	62ff	1_10, 3_11

Table 10-2: Rule base for aeronautics patents

Concepts	
glider	:= <glider,1.0,kite,1.0>
balloon	:= <balloon,1.0>
airship	:= <airship,1.0>
helicopter	:= <helicopter,1.0>
rocket	:= <rocket,1.0,missile,1.0>
rotor	:= <rotor,1.0>
steering_device	:= <rudder,1.0,steering,0.8>
payload	:= <payload,1.0, passenger, 1.0, freight,1.0, cargo,1.0, luggage,0.8, weapons,1.0,missiles,1.0>
evacuation	:= <evacuation,1.0,escape,1.0,rescue,1.0>
inflatable	:= <inflatable,1.0>
slide	:= <slide,1.0, ramp,1.0, slideway,1.0, chute,1.0, system,0.8, device,0.8>
safety	:= <safety,1.0>
escape	:= <escape,1.0,retreat,0.8>
catapult	:= <catapult,1.0>
launch	:= <launch,1.0,take-off,1.0>
aircraft	:= <aircraft,1.0>
light	:= <light,1.0,lightweight,1.0,ultralight,1.0>
parachute	:= <parachute,1.0,parachute-wing,1.0>
propulsion	:= <propulsion,1.0>
motor	:= <motor,1.0,engine,1.0,turbine,1.0>
tip	:= <tip,1.0>
fin	:= <fin,1.0,sail,0.8>
vertical	:= <vertical,1.0>
wing	:= <wing,1.0>
control	:= <control,1.0>
Rule Type \mathcal{R}_1	
1_01	<balloon> 0.3
1_02	<airship> 0.5
1_03	<helicopter> 0.4
1_04	<rotor> 0.1
1_05	<rocket> 0.5
1_07	<glider> 0.7
1_08	<parachute> 0.5
1_09	<launch> 0.1
1_10	<propulsion> 0.1
Rule Type \mathcal{R}_2	
2_01	<balloon> 0.1 <airship> 0.1
2_02	<helicopter> 0.1 <rotor> 0.1
2_03	<glider> 0.2 <parachute> 0.2
Rule Type \mathcal{R}_3	
3_01	<evacuation,slide> 0.1 <evacuation,slide> 0.1 * *
3_02	<inflatable,slide> 0.1 <inflatable,slide> 0.1 * *
3_03	<inflatable,slide> 0.1 <evacuation,slide> 0.1 * *
3_04	<evacuation,slide> 0.1 <safety> 0.1 * *
3_05	<evacuation,slide> 0.1 <escape> 0.1 * *
3_06	<catapult,launch> 0.4 <catapult,launch> 0.4 * *
3_07	<light,aircraft> 0.5 <light,aircraft> 0.5 * *
3_08	<glider> 0.4 <light,aircraft> 0.5 * *
3_09	<light,aircraft> 0.2 <balloon> 0.2 * *
3_10	<light,aircraft> 0.2 <airship> 0.2 * *
3_11	<motor,rotor> 0.2 <motor,rotor> 0.2 * *
3_12	<vertical,fin> 0.3 <vertical,fin> 0.3 * *
3_13	<tip,fin> 0.3 <tip,fin> 0.3 * *
3_14	<vertical,fin> 0.3 <tip,fin> 0.3 * *
3_15	<wing,tip> 0.2 <vertical,fin> 0.2 * *
3_16	<aircraft,control> 0.2 <aircraft,control> 0.2 * *

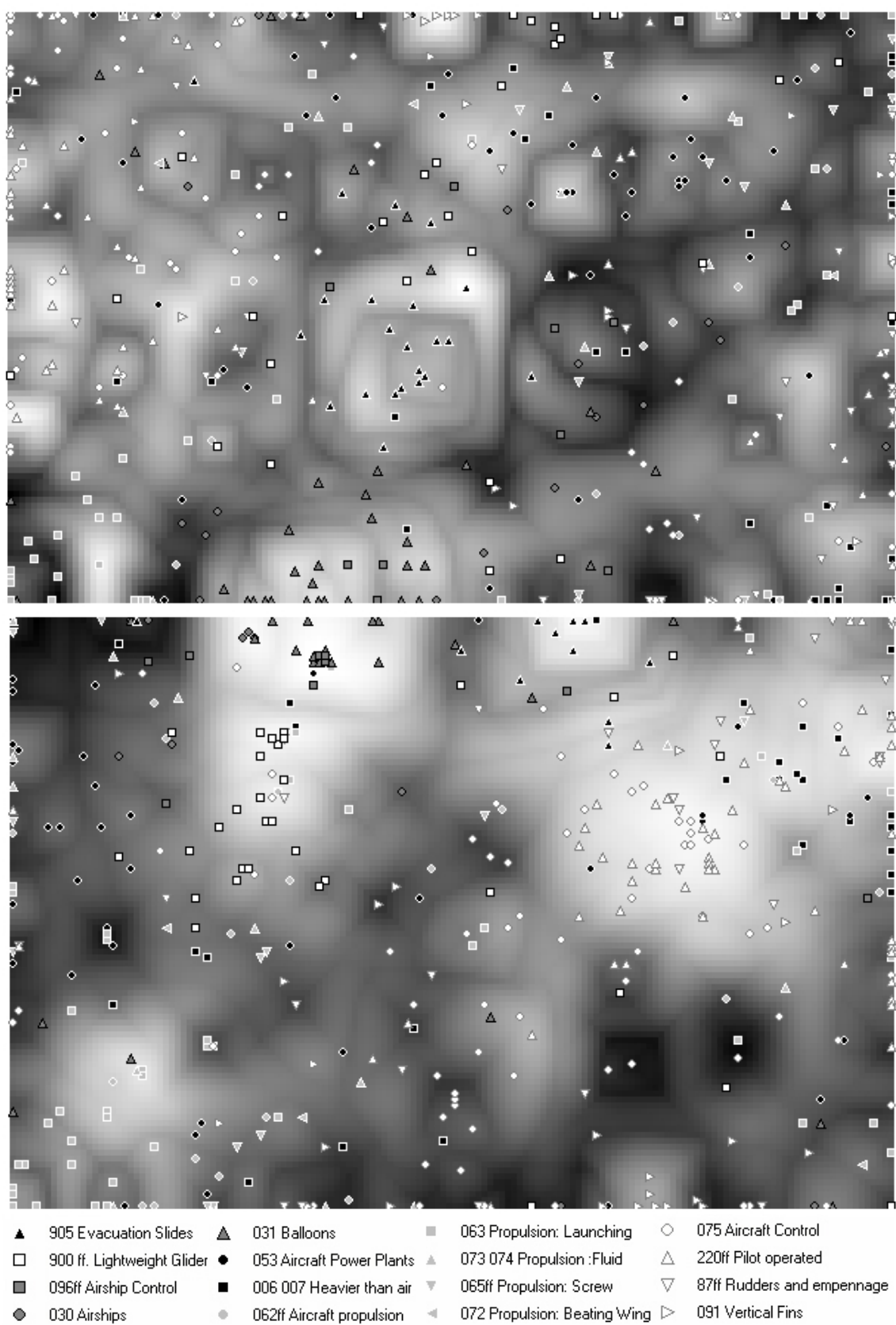


Figure 10-16: Maps of aeronautics patents: (a) pre-structured, (b) refined map, (c) legend of symbols

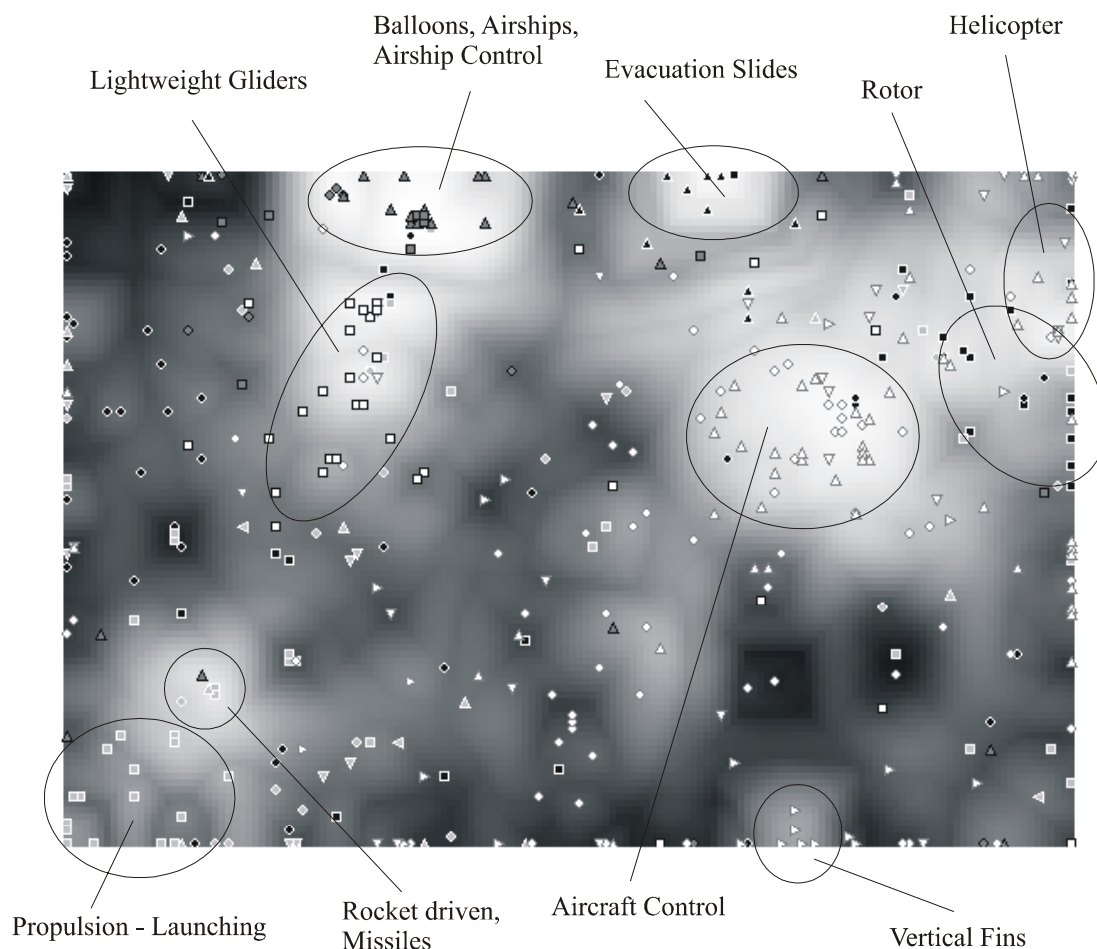


Figure 10-17: Refined map of aeronautics patents with additional inscription

10.10.2 Experiment II: Airplane Descriptions

This scenario deals with a relatively small collection of airplane descriptions. The aim of this experiment is to demonstrate and evaluate the use of attribute-value pairs (cf. chapter 10.6.1) within the semantic refinement module. The collection consists of 44 documents describing different aircraft available from Boeing and has been downloaded directly from the Boeing website¹¹. Each document describes a different type of aircraft and contains various technical information in plain text, like the number of passengers, payload, maximum velocity. Again, a basic classification of the documents in different categories is available: commercial aircraft, military aircraft and rotor based aircraft.

Figure 10-18 shows the pre-structured document map of this collection which has been computed with the statistical document analysis module (cf. chapter 6.2.1.3) without a semantic refinement. Details on collection statistics and parameters for map generation can be found in table 10-5 (p. 244). The map shows a clear differentiation between the three pre-defined classes. Related aircraft can be found closely together within the groups, like the 757, 767 and 777 or the different C17 and B-52 models. The helicopter group is recognizable as well with the V-22 lying away from the rest of the group, which is semantically justified since the V-22 is not a real helicopter but a hybrid: “*The V-22 is a tiltrotor aircraft, taking off and landing*

¹¹ www.boeing.com as of May 2000

like a helicopter, but, once airborne, its engine nacelles can be rotated to convert the aircraft to a turboprop airplane capable of high-speed, high-altitude flight.”

The document structure shown in figure 10-18 is based on the word distribution in the documents. As the map reflects, a comparison based on this statistical information provides a meaningful insight to the collection’s structure. However, certain aspects cannot be taken into consideration here. For example, it is not possible to have the structure reflect information on how many passengers or how much weight the planes are able to carry. Though, in a knowledge management context this might be an important aspect of the data that the analyzing user is interested in. This is where the proposed rule-based approach comes in. In particular, attribute-value pairs can be used within the rules in order to achieve a weighting of certain

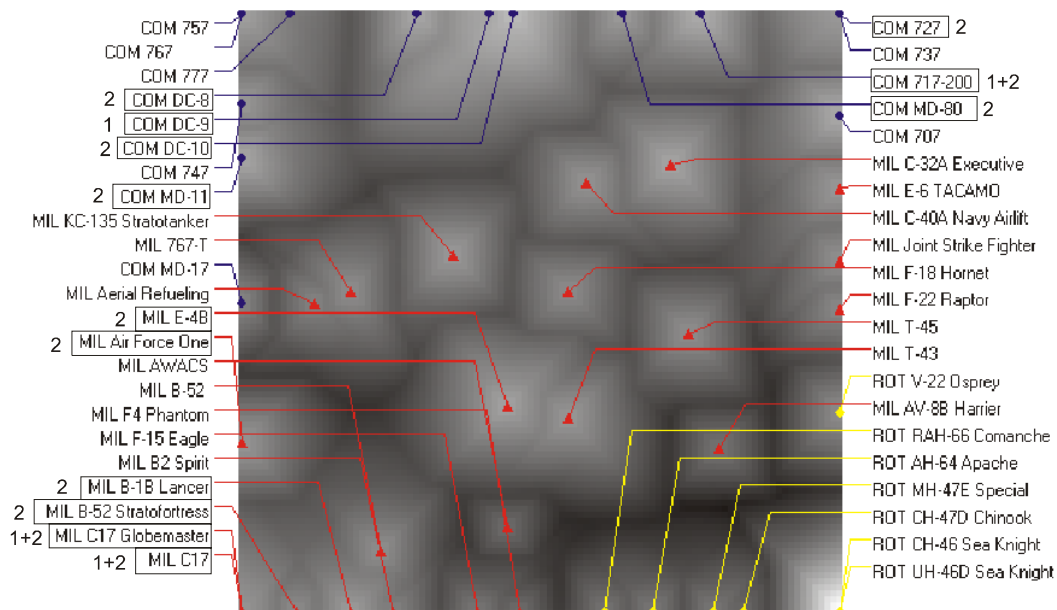


Figure 10-18: Collection of aircraft descriptions without semantic refinement. The type names of the aircraft described in the respective documents are indicated, along with a classification in commercial (COM), military (MIL) and rotor based (ROT) aircraft.

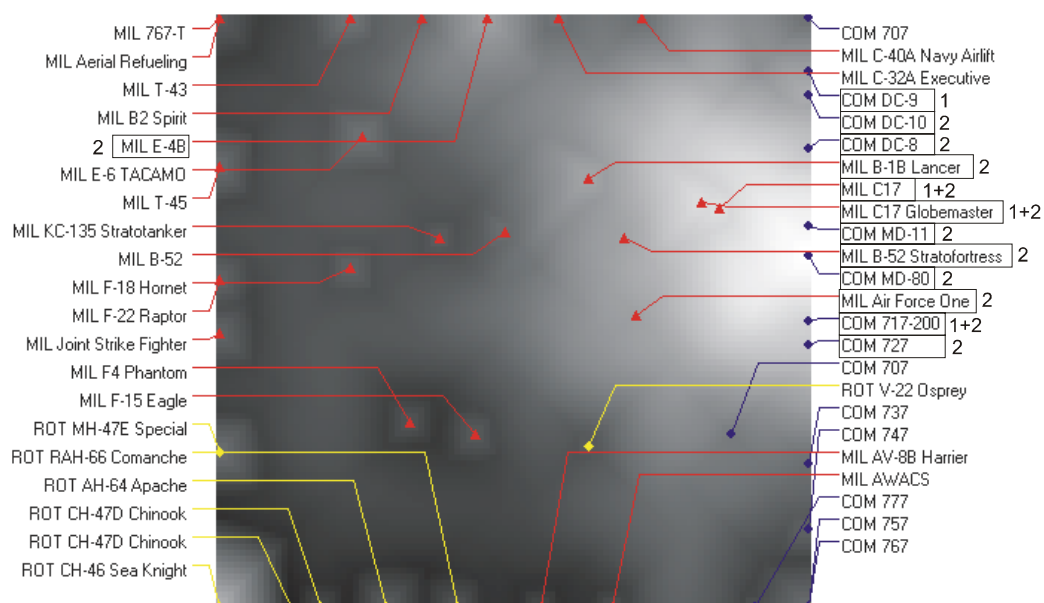


Figure 10-19: Collection of aircraft descriptions with focus on large wingspan and large weight

technical parameters.

A small and simple rule base is shown in table 10-3. It defines two attributes describing aircraft features that the analyst may be interested in: wingspan and maximum take-off weight or payload. In both cases, the ranges of values which interpret linguistic terms are defined as vague intervals. The regular expression set for the syntactic matching will match syntactic constructs like “*large wingspan*”, “*wingspan is very large*” or “*wingspan of 90 feet*”. The ‘weight’ rule uses a simple heuristic: The concept ‘weight’ contains the term ‘pound’ in order to match laxly phrased sentences like “*The MD-82’s and the MD-88’s are 149,500 pounds*”,

Table 10-3: Rule base for stressing attribute-value pairs related to wingspan and weight in airplane descriptions

Concepts							
wingspan	:= <wingspan,1.0>						
weight	:= <weight,1.0,pounds,1.0>						
Attributes							
wingspan				weight			
0 1000				0 1000000			
small	0 5 25 30			small	0 500 10000 15000		
medium	25 35 70 90			medium	15000 30000 60000 90000		
large	80 90 200 220			large	70000 80000 900000 1000000		
Syntactic Regular Expressions for Attribute-Value Pairs							
AVP							
/&1/&2							
/&2/VB [DGNPZ] [(/RB)] */&1							
/&2/IN/&1							
Rules (Rule Type \mathcal{R}_1)							
No. 1:	<large wingspan>	1.0					
No. 2:	<large weight>	1.0					

Table 10-4: Numeric information regarding wingspan and weight contained in aircraft description documents

aircraft	wingspan	max. weight or payload
717-200	(*) 93.3 feet	(*) 114,000 pounds
727		(*) 210,000 pounds
DC-10		(*) 175,000 pounds
DC-8		(*) 116,000 pounds
DC-9	(*) 93.3 feet	
MD-11		(*) 602,555 pounds
MD-80		(*) 160,000 pounds
Air Force One		(*) 833,000 pounds
B-1B Lancer		(*) 477,000 pounds
B-52 Stratofortress		(*) 488,000 pounds
C17	(*) 169.75 feet	(*) 585,000 pounds
C17 Globemaster	(*) 169.75 feet	(*) 585,000 pounds
E-4B		(*) 800,000 pounds
F-18 Hornet		13,700 pounds
F4 Phantom		16,000 pounds
Joint Strike Fighter		18,000 pounds
T-45	30.10 feet	13,636 pounds
AH-64 Apache		16,000 pounds
CH-47D Chinook		25,000 pounds
V-22 Osprey		20,000 pounds

which refers to the take-off weight. Note that the collection used in this example is not strictly standardized: Not all documents describe all features of the different aircraft in the text (often there is an external table with the relevant data which has not been taken into consideration due to the scope of this experiment). In fact, the payload or maximum take-off weight is the only feature verbally described within a majority of the documents. Therefore it is not possible to reasonably define further rules that show different structural analysis aspects.

Figure 10-19 shows the result of the semantic refinement (details on parameters can be found in table 10-5). The documents that have triggered a rule are indicated by a box surrounding the document name, along with the number of the triggering rule (to ease the comparison of pre-structured and refined map the same information is given in figure 10-18). Table 10-4 shows the attributes for which rules have been defined and their numeric value as manually found in the documents. The table only contains those documents that actually contain information on the examined feature. In the table a ‘(*)’ denotes a successful triggering of rules for the respective document. A very high precision of rule triggering can be observed. This particularly holds for the heuristic ‘weight’ rule – a fact that is partly due to the highly special-

Table 10-5: Parameters and statistics of collections and maps. Concerning the given computation times note that all experiments have been performed on a Pentium II 350 MHz machine with 128 MB RAM and WinNT 4.0.

collection	name	aeronautics patent abstracts	Boeing airplane descriptions
	# documents	496	44
	avg. # words	121	557
indexing	size of vocabulary	2,669	2445
	avg. # indexing terms / document	33	172
	weighting scheme	TF	TF
	similarity measure	Euclidean	Euclidean
doc. space	dimension	40	40
	avg. stress	0.63	0.019
semantic refinement	movement length factor (p)	0.1	0.08
	# iterations (t_{\max})	100	10
	# rules	28	2
	# resulting attractors	10,737	69
SOM	size of SOM	150×100	100×100
	training steps per document	20	20
times	basic map generation	15 min 8 sec	1 min 3 sec
	semantic refinement	8 min	4 sec

ized collection: There are simply no weights in the considered range that would not talk about maximal take-off weight or payload. Otherwise a certain amount of mismatching would have to be expected.

Considering the ‘biased’ map in figure 10-19, one can also clearly observe the effect of superimposing a ‘view’ on the pre-structured map: Commercial and military aircraft with large wingspan and heavy weight, respectively, have moved closer together and form a group in a bright-shaded area. At the same time matching commercial aircraft are still close to the remaining commercial plane descriptions and military aircraft with high weight and wingspan still show a connection to the group of the other military planes. This information results from the overall similarity of the documents rather than from the explicitly defined rules.

As the example shows, attribute-value pairs allow to compare (vague) values that would have been impossible to compare in the simple term-vector based approach. However, due to the variability of natural language it can sometimes be hard to correctly identify an attribute-value pair with the proposed approach. At this point there is also no unit conversion involved when comparing different numeric values. Thus, there is a lot of room for improvement in the actual attribute-value pair comparison approach. However, this experiment promises that the general idea provides a meaningful contribution to the structural analysis.

10.10.3 Conclusion

The experiments indicate that the semantic refinement module is able to successfully improve and refine the displayed structure of a map with respect to a user-defined ‘bias’. The first experiment has shown that the structure of a basic map can be meaningfully influenced by a rather simple set of rules. The resulting map reflects the bias, and a microanalysis of the structures suggests that the grouping of documents can be understood by considering the general document similarity and the relationships established in the user-defined rule base. The second experiment has shown that expressive attribute-value-rules can be used for effectively establishing semantically rich relationships. All in all these are promising results. However, a broader evaluation in the context of practically oriented case studies is necessary to assess the value, usefulness and usability of the proposed extension in full.

11 Conclusion

11.1 Summary of Results and Contributions

An important technological aspect of knowledge management is to support companies and organizations in their efforts to structure, to condensate, and to learn from their documented knowledge assets. Since a great deal of corporate or organizational knowledge is contained in textual documents, techniques which allow a task-adequate access to text collections play an important role. Visualizing the semantic structure of text corpora promises a means for easily identifying outliers, associations and clusters in the document space and thus may help to analyze collections of documents and assist analysts in their efforts to effectively exploit textual knowledge assets. While the usefulness of graphical collection overviews for standard search tasks still has to be proven, this work has contributed to a better understanding of the value of a document map technology for supporting text corpus analysis tasks in knowledge management. The task model in chapter 2 provides a means for describing tasks which are relevant for analyzing text corpora in the context of knowledge management. As a domain-specific taxonomy for characterizing, comparing and typifying real-world analysis tasks, it is a contribution of its own and can be applied for various purposes, such as requirement analysis or system classification and design. In this work, the task model served as a guide for the technological development and as a yard stick for evaluation.

Following an extensive critical review of techniques for structuring and visualizing document collections and their basics (chapters 3 – 5), which was lead by the motivation to support corpus analysis tasks, a novel document map approach was proposed: The flexible basic method (chapter 6) allows a fine-granular analysis of specialized document collections. It can be tailored to domain-specific needs since the module for assessing the similarity of documents is exchangeable. An extension of this approach (chapter 10) allows to incorporate a personal ‘bias’ into the map generation process while the flexibility of the basic framework is preserved. The map-centered corpus analysis system DocMINER (chapter 7) realizes these methods. It tightly integrates powerful tools for explorative and goal-directed collection access with the map display and thereby completes the technological contributions of this work.

The basic method was thoroughly investigated in experiments, case studies and a comparative laboratory study. It turned out that the document map approach offers meaningful insights into a collections’ structure and allows to effectively study relationships between single documents and document groups. As the qualitative (chapter 8) and quantitative (chapter 9) investigations have shown, it is particularly successful for supporting tasks that require a detailed structural analysis of document-document, document-topic or document-specification relationships. The overall approach can handle moderately sized document collections (up to 10,000 documents have been processed in experiments so far). However, the visualization approach seems to be most powerful for several hundred items to be displayed – an order of magnitude which is practically relevant as the case studies and the survey in knowledge-intensive industries have shown. Besides a detailed evaluation of their effectiveness, fruitful ways of applying document maps for important corpus analysis tasks have been worked out:

The case studies could provide experience reports which can be of use for adapting document map applications to further domains.

11.2 A Media-Theoretic Perspective on Document Maps

The idea of providing visual access to text corpora for knowledge management can also be discussed from the viewpoint of cultural sciences, in particular from a media-theoretic perspective: Document maps can be seen as description instruments which transcribe one representation form of symbol systems into a different description language and thereby generate previously unavailable *readability*. According to Jäger [Jäg01] there are two particularly interesting approaches of semantic ratification: The first (and commonly known) is an *intramedial* characteristic of natural languages, namely the possibility to communicate about language by language. This quality of language allows partners in a communication to define the meaning of linguistic statements by paraphrasing, explanation and explication (cf. Watzlawick's psychological concept of meta-communication, [WBJ90]). The second approach is an *inter-medial* method which uses further communication systems (e.g. document maps) for translating, commenting, explaining or making explicit aspects of the first system's semantics (e.g. the similarity structure a corpus of corporate documents).

In both cases the meaning of (parts of) symbol systems can be opened up – while at the same time a new quality of semantics comes into play. Jäger calls both ratification approaches *transcriptive* [Jäg01] and illustrates the concept of *transcription* by the example of externalizing implicit knowledge (cf. section 1.1): Externalization is not only a mapping of mental concepts (tacit, implicit knowledge) to explicitly formulated rules. Rather, the transcriptive formulation of rules throws new light on the mental concepts (which were originally hard or even impossible to communicate) since they are now accessible in a new form, gaining a new status and quality. It is this transcription that allows new discourses about the implicit knowledge: rules are *readable* and can be critically discussed and reflected with respect to the implicit concepts they derive from – a quality which was originally not available.

A visual presentation of a text collection's similarity structure is thus a classic application of transcription: The structure is transcribed into a new symbol system (the map) which not only *represents* the existing similarity structure but offers a new semantic quality and a readability which was previously unavailable. The analysts can start discourses on the corporate document corpus (such as critically reflecting its contents in certain respects) in a way which would not have been possible before. As an example recall the case study from section 8.2: Technical writers could gain insight in the structures of their own text material and were enabled to discuss and assess the semantic structure of user manuals in a goal-directed way. This kind of goal-directed readability was generated by the transcriptive effect of document maps.

11.3 Outlook

Graphical collection overview technology can effectively support practically relevant corpus analysis tasks. This hypothesis is supported by the encouraging results achieved during the evaluation of the basic document map approach and the promising results of the first experiments with the additional adaptability component. Thus, it is very promising to further press ahead research on these issues. Examples of interesting research questions are given in the following sections.

11.3.1 Extending the Evaluation

Application-oriented evaluation of the semantic refinement module: The semantic refinement component as introduced in chapter 10 provides a means for adaptability. It enables the analyst to incorporate a personal bias into the map generation process via a user-defined set of rules. Though first experiments have shown promising results, a deeper evaluation in an application-oriented way is necessary. Due to the time-line of development the method could not be considered in case studies. Presentations of the method's idea to practitioners yielded positive feedback, yet its applicability and usefulness for practical problems is still open to a great extent. Questions which have to be addressed in future evaluations include: Do users understand the 'double' semantics of the approach in real application contexts, i.e. the combination of document similarity computed by the document analysis module and the rule-defined degree of relatedness (see also section 11.3.2)? Are they able to successfully model rules that express a desired bias? Does the approach indeed add value to the practical analysis of specialized document collections?

Comparative evaluation of different text-access paradigms: Though the understanding of graphical overviews for text corpus analysis could be improved by this thesis, the achieved results are only the tip of the iceberg. Further evaluations are necessary to get a broader picture of the usefulness and applicability of document maps and related techniques for this application domain. An interesting issue is to compare the effectiveness of the proposed document map approach with other graphical overview methods (cf. chapter 5) and alternative text-access paradigms, such as scatter-gather browsing (cf. section 5.2.1). This should be done not only in laboratory settings but also in field studies. For example, the support offered by document maps for corpus analysis in terminology work (cf. section 8.3) could be compared with the support offered by alternative text-access paradigms and statistical evaluations of term usage. The collaborative research center "Media and Cultural Communication" (cf. section 8.3.1.1), for instance, provides a suitable context for this kind of research.

11.3.2 Possible Improvements of Map Technology and System

Enhancing the interpretability of the visualization: A property of the SOM mapping method is that documents or groups of documents that are neighbored in the map display are also neighbored in the semantic document space (and thus similar). However, there may be neighbored (similar) clusters of documents in the semantic document space which are located in some distance in the map display (cf. section 6.3). Stretching of some kind is a general and inevitable problem of visualizing inherently high-dimensional object arrangements. But since this effect may be a source of misinterpretation it is desirable to provide additional interpretation aids. In fact, the SOM approach offers a rich basis for visualizing complex relationships: While the chosen *P*-matrix visualization method presents the cluster structure of the input space (cf. section 6.2.3), the similarity coloring method (cf. section 4.2.3) indicates the areas in the map which are neighbored in the input space by assigning similar hues to neighbored clusters. Thus, by using a suitable overlay technique the desired interpretation aid can be realized: On demand, the hues of the similarity coloring could be superimposed on the gray scales of the *P*-matrix visualization.

Document maps in cooperative environments: The case studies have shown that corpus analysis often contains cooperative aspects: For example, structuring use cases in software engineering (chapter 8.1) and publication analysis for terminology work (chapter 8.3) involve groups of analysts who collaboratively have to understand and analyze a collection's inherent structure. Presently, the DocMINER system offers only limited support for cooperative work.

Consequently, interesting improvements involve the development of adequate cooperative system functions (such as extended annotation and discussion utilities) and the integration of DocMINER into a groupware system which provides a suitable infrastructure for distributed cooperative work (such as a virtual collaborative document map workspace).

Enhancing and extending the semantic refinement component: The semantic refinement module allows to incorporate a bias into the map generation process via a user-defined set of rules. It follows the principle of inclusivity (cf. section 5.8.4), i.e. it considers both, the document relationships determined by the semantic analysis module in the pre-structuring step as well as the degree of relatedness computed by applying the rules, cf. chapter 10.1). In order to help the analyst to better understand the resulting structures in the map, the integration of an ‘explanation component’ into the system will surely be of use: For a selected set of documents in the map workspace all relevant information could be displayed, including the original document distance, rules that have been triggered by the documents, the degree of contribution to the document movement of single rules, or a visualization of the total attractor strength between the documents.

Alternative application of the semantic refinement module: An alternative application of the semantic refinement module is its use for realizing an exclusive view on the collection (principle of exclusivity, cf. section 5.8.4). Then, only the relationships defined in the rule base would be considered, leading to a more goal-directed analysis of a collection’s structure. Such an application requires the definition of a similarity measure on basis of the rule approach. While some of the proposed rule types could directly contribute to such a measure, other rule types would have to be adapted. Though this aspect was not in the focus of interest in this work, it is an interesting and important application which is worthwhile to be considered. The proposed rule base approach may offer a rich starting point for such a development.

11.3.3 Further Research Directions

Connecting textual and numerical data for generating document maps: A recent trend in text-mining is to integrate textual as well as numerical information into knowledge discovery processes (for example in finance data analysis). Thus, connecting textual and external numerical data for generating document maps is an interesting research issue: How can structured numerical information, possibly from external sources, adequately be integrated into the semantic analysis of textual documents? Since the basic framework (chapter 6) allows to choose adequate document analysis modules for the considered application domain, this question mainly concerns the development of new, powerful document analysis methods, based on results of information extraction, data mining and case-based reasoning. Methods of the latter research field in principle allow the incorporation of non-textual information into the case (or document) analysis process (cf. [Lenz98b]). A correspondingly adapted document map system should then comprise additional functions for understanding and analyzing textual and numerical relationships.

Tightly integrating textual and graphical information in map displays: The comparative empirical study (cf. section 9.5.1.1) and discussions in literature (cf. [Hea99], p. 274 ff.) suggest that users of graphical corpus overviews would like to see more text, tightly coupled to the graphical display or another expressive arrangement of text clusters. Consequently, an important research issue is find out how an effective combination of textual and graphical overviews should ideally look like and how it could be realized technically. Possible combinations could e.g. use textual cluster representations (like in scatter-gather browsing, cf. section 5.2.1), connect these representations by symbols and color-coding to the graphical display, use graph representations of textual information in order to display the relatedness of

play, use graph representations of textual information in order to display the relatedness of corresponding documents (for example for document map areas in which the user is currently zoomed in), and generate more powerful textual field descriptions. The latter line of development could benefit from results of automatic text summarization, or linguistic results regarding text reception for the considered genres. A more detailed investigation of this direction may also be of use for improving graphical information retrieval tools.

References

- [AaNy95] Aamodt, Agnar, and Mads Nygard. Different roles and mutual dependencies of data, information, and knowledge – an AI perspective on their integration. *Data and Knowledge Engineering* 16, Elsevier, 1995, pp. 191-222
- [Allan95] Allan, James: Automatic Hypertext Construction, PhD Thesis, Cornell University, 1995
- [Allan96] Allan, James: Automatic Hypertext Link Typing. *Proceedings of Hypertext '96*, Washington, USA, 1996
- [AlMi90] Alberico, Ralph and Mary Micco: Expert Systems for Reference and Information Retrieval. Meckler, Westport, Conn., 1990
- [ALS97] Allan, James, Leouski, Anton V., Swan, Russell C.: Interactive Cluster Visualization for Information Retrieval. Tech. Rep. IR-116, Center for Intelligent Information Retrieval, University of Massachusetts, 1997
- [Amma99] Amman, Margret: Information Design – Grundlage eines Dokumentationskonzepts. Tagungsband der tekom-Frühjahrstagung 1999 in Innsbruck, Gesellschaft für technische Kommunikation e.V., Stuttgart, 1999
- [BAB+97] Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkil, S., Trevor, J. and Woetzel, G.: Basic Support for Cooperative Work on the World Wide Web, *International Journal of Human-Computer Studies* 46(6): Special issue on Innovative Applications of the World Wide Web, 1997
- [BaPa92] Bauer, H.-U., Pawelzik, K.: Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, 1992
- [Bat89] Bates, M.: A 'berrypicking' model of information retrieval. *Online Review* 13(5):408–424, 1989
- [BCGH+99] Booker, A., Condliiff, M., Greaves, M., Holt, F., Kao, A., Pierce, D., Poteet, S., Jason Wu, Y.: Visualizing Text Data Sets. *IEEE Computing in Science & Engineering*, July/August 1999, pp. 26–35
- [BCL+94] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen H.F., Secret, A. The World-Wide Web. *Communications of the ACM*, 37(8), August 1994, pp. 76–82
- [BeHo00] Becks, A., Host, M. Visuell gestütztes Wissensmanagement mit Dokumentenlandkarten. *Wissensmanagement – Das Magazin für Führungskräfte*, Heft 4, Juli 2000, S. 33–36
- [BeHo99] Becks, A., Host, M. Qualitätsprüfung mit Dokumentenlandkarten. Tagungsband der tekom-Frühjahrstagung 1999 in Innsbruck, Gesellschaft für technische Kommunikation e.V., Stuttgart, 1999
- [BeKo99] Becks, Andreas, Köller, Jörg. Automatically Structuring Textual Requirement Scenarios. In: *Proceedings of the 14th IEEE Int. Conf. on Automated Software Engineering*, Cocoa Beach, Florida, USA, 1999 (also appeared as Crews Report 99-15, RWTH Aachen)

-
- [BeSe01] Becks, Andreas, and Christian Seeling. A Task Model for Text Corpus Analysis in Knowledge Management. In: UM2001 – 8th International Conference on User Modeling, Workshop on User Modeling, Machine Learning and Information Retrieval, Sonthofen, Germany, July 2001
 - [BFW98] Brown, Mike, Christiane Förtsch and Dieter Wißmann. Feature Extraction – The Bridge from Case-Based reasoning to Information Retrieval. In Gierl, Lenz (eds.): Proceedings of the 6th German Workshop on Case-Based Reasoning, IMIB Report, University of Rostock, 1998
 - [BHK95] Burke, Robin, Hammond, Kristian, Kozlovsky, Julia. Knowledge-based Information Retrieval from Semi-Structured Text. In: Working Notes from AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval. American Association for Artificial Intelligence, 1995, pp. 19-24
 - [BHMP92] Blosseville, M.J., Hébrail, G., Monteil, M.G., Pénot, N. Automatic Document Classification: Natural Language Processing, Statistical Analysis, and Expert System Techniques used together. In: Proceedings of the 15th ACM SIGIR Conference on Research and Development in Information Retrieval, Denmark, 1992, pp. 51-57
 - [BK01] Becks, Andreas und Ralf Klamma. Kooperative Dokumentanalyse in einem interdisziplinären Forschungskolleg. In: H.-P. Schnurr, S. Staab, R. Studer, G. Stumme. Y. Sure (Hrsg.): Beiträge der 1. Konferenz “Professionelles Wissensmanagement - Erfahrungen und Visionen”, Baden-Baden, 14.-16. März 2001, S.289-307
 - [BKJJ97] v.Buol, B., S. Kethers, M. Jeusfeld, M. Jarke. A Terminology Server for Cooperative Terminology Work. 2nd IFCIS Conference on Cooperative Information Systems CoopIS'97, Kiawah Island, South Carolina, 1997
 - [Blah87] Blahut, Richard E. Principles and Practice of Information Theory. Addison-Wesley, Reading, Massachusetts, 1987
 - [BoFa94] Bonjour, M., Falquet, G.: Concept Bases: A Support to Information Systems Integration. In: Brinkkemper, S., Wassermann, T., Wijers, G. (Hg.): Proceedings of the 6th Conference on Advanced Information Systems Engineering (CAiSE'94), Utrecht, 6.-10.6.1994, Lecture Notes on Computer Science, No. 811, Springer Verlag, 1994
 - [BöKr99] Böhmann, Tilo und Helmut Krcmar. Werkzeuge für das Wissensmanagement. In: Conny H. Antoni, Tom Sommerlatte (Hrsg.): Report Wissensmanagement – Wie deutsche Firmen ihr Wissen profitabel machen, Symposion Publishing, 1999
 - [BoPi95] Bosc, P., Pivert, O. SQLf: A Relational Database Language for Fuzzy Querying. IEEE Transactions on Fuzzy Systems, Vol 3(1), 1995
 - [Bota93] Botafogo, Rodrigo A. Cluster Analysis for Hypertext Systems. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA, USA, June, 1993
 - [Brae99] Brändle, Max. Die elektronische Dokumentation – Stand der Technik und Trends. Tagungsband der tekom-Frühjahrstagung 1999 in Innsbruck, Gesellschaft für technische Kommunikation e.V., Stuttgart, 1999
 - [Brau95] Brause, Rüdiger. Neuronale Netze, B.G. Teubner, Stuttgart, 1995
 - [Bri92] Brill, E. A simple rule-based part of speech tagger. Proceedings of the Third Conference on Applied Natural Language Processing, 1992
 - [Bri94] Brill, E. Some advances in transformation-based part of speech tagging. Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, USA, 1994
 - [Bri95] Brill, Eric. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics, 21(4), December 1995
-

- [BrJo95] Brice A, Johns W R.: Open Process Simulation. OO-CAPE report. QuantiSci report IC4381-2, 1995
- [BrSc85] Brachmann, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171-216, 1985
- [BSJ00a] Becks, A., Sklorz, S., Jarke, M.: A Modular Approach for Exploring the Semantic Structure of Technical Document Collections. *Proceedings of AVI 2000, Int. Working Conference on Advanced Visual Interfaces*, Palermo, Italy, 2000, pp. 298–301
- [BSJ00b] Becks, A., Sklorz, S., Jarke, M.: Exploring the Semantic Structure of Technical Document Collections: A Cooperative Systems Approach. In Etzion, O., Scheuermann, P. (Eds.): *Cooperative Information Systems, Proceedings of the 7th Int. Conference on Cooperative Information Systems*, LNCS 1901, Eilat, Israel, 2000, pp. 120–125
- [BST94] Belkin, Nicholas J., Adelheit Stein, Ulrich Thiel. Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems. *Arbeitspapiere der GMD No. 875*, Sankt Augustin, 1994
- [BST98] Becks, A., Sklorz, S., Tresp, C. Semantic Structuring and Visual Querying of Document Abstracts in Digital Libraries. In *Proc. of the Second European Conference on Research and Advanced Technology for Digital Libraries (LNCS 1513)*, Crete, Greece, 1998, 443-458
- [Buol99] v. Buol, B.: *Qualitätsgestützte, kooperative Terminologearbeit*. Dissertation, RWTH Aachen, 1999
- [CDMR98] Crossley, Martin, Davies, John, McGrath, Andrew, Reijman-Greene, Marek. The Knowledge Garden. In: *Proceedings of the 2nd International Conference on Practical Aspects of Knowledge Management (PAKM98)*, Basel, Switzerland, 1998
- [Chal93] Chalmers, Matthew. Using a Landscape Metaphor to Represent a Corpus of Documents. In: Andrew U. Frank, Irene Campari (Eds.): *Spatial Information Theory – A Theoretical Basis for GIS*, *Lecture Notes in Computer Science*, LNCS 716, 1993, pp.377–390
- [ChCh92] Chalmers, Matthew, Chitson, Paul. Bead: Explorations in Information Visualization. In: *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, 1992, pp. 330-337
- [Chen94] Chen, Hsinchun. Collaborative Systems: Solving the Vocabulary Problem. *IEEE Computer*, May, 1994
- [Chen97] Chen, Chaomei. Structuring and Visualising the WWW by Generalised Similarity Analysis. *Proceedings of ACM Hypertext 97*, Southampton, UK, 1997
- [CHO+94] Chen, H., Hsu, P., Orwig, R, Hoopes, L., Nunamaker, J.F. Automatic Concept Classification of Text from Electronic Meetings. *Communications of the ACM*, 37(10), 1994, pp. 56-73
- [CHSS98] Chen, H., Houston, A.L., Sewell, R.R., Schatz, B.R. Internet Browsing and Searching: User Evaluations of Category Map and Concept Space Techniques. *Journal of the American Society for Information Science*, 49 (7), 1998, pp. 582–603
- [CKP93] Cutting, D.R., Karger, D.R., Pedersen, J.O. Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, PA, USA, June, 1993
- [CKPT92] Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, Denmark, June, 1992

-
- [CoDo86] Cooke, N.M., and J.E. McDonald. The application of psychological scaling techniques to knowledge elicitation for knowledge-based systems. In J.H. Boose, B.R. Gaines (eds.): Knowledge-Based Systems: Knowledge Acquisition Tools for Expert Systems, Vol. 2, Academic Press, New York, 1986
 - [CR97] The full computing reviews classification system. *Computing Reviews*, 37(1), 4–16, 1997
 - [CSO96] Chen, Hsinchun, Schuffels, Chris, Orwig, Richard. Internet Categorization and Search: A Self-Organizing Approach. *Journal of Visual Communication and Image Representation*, Vol. 7, No. 1, pp. 88-102, 1996
 - [Davi83] Davidson, Mark L. *Multidimensional Scaling*. John Wiley & Sons, 1983
 - [DDH90] Deerwester, Scott, Dumais, Susan T., Harshman, Richard. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6), 1990, pp. 391-407
 - [DHJ+98] Davidson, George S., Hendrickson, Bruce, Johnson, David K., Meyers, Charles E., Wylie, Brian N. Knowledge Mining With VxInside: Discovery Through Interaction. In: *Journal of Intelligent Information Systems*, Vol. 11, No. 3, 1998, pp.259-285
 - [Dieb95] Dieberger, Andreas. Providing Spatial Navigation for the World Wide Web. *Spatial Information Theory, Proceedings of COSIT'95*, Semmering, Austria, LNCS 988, Springer, 1995
 - [DIN77] DIN 820: Blatt 1: Normungsarbeit: Grundsätze. Deutsches Institut für Normung, Beuth Verlag, Berlin, 1977
 - [DIN86] DIN 2339/1: Ausarbeitung und Gestaltung von Veröffentlichungen mit terminologischen Strukturen. Teil 1: Stufen der Terminologearbeit. Deutsches Institut für Normung, Beuth Verlag, Berlin, 1986
 - [DMRA97] Delcambre, Lois M.L., Maier, David, Reddy, Radhika, Anderson, Lougie. Structured Maps: modeling explicit semantics over a universe of information. *International Journal on Digital Libraries*, Vol. 1, No. 1, 1997, pp. 20-35
 - [Doyle61] Doyle, Lauren B. Semantic Road Maps for Literature Searchers. *Journal of ACM*, 8, 1961, pp. 553-578
 - [Dubin95] Dubin, David. Document Analysis for Visualization. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, USA, July, 1995
 - [DuPr80] Dubois, Didier, Prade, Henri. New results about properties and semantics of fuzzy set-theoretic operators. In: Wang, P.P and Chang, S.K. (eds.): *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, pp. 59–75, Plenum Press, New York, 1980
 - [DuPr85] Dubois, Didier, Prade, Henri. A review of fuzzy set aggregation connectives. *Information Sciences*, 36, pp. 85–121, 1985
 - [DWRM96] Davies, J., Weeks, R., Revett, M., McGrath, A. Using Clustering in a WWW Information Agent. In: *Proc. of the 18th BCS Information Retrieval Colloquium*, British Computer Society, 1996
 - [Ead84] Eades, P. A heuristic for graph drawing. *Congressus Numerantium*, 42:149-160, 1984
 - [FaLi95] Faloutsos, Christos, Lin, King-Ip. FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In: *Proceedings of the ACM SIGMOD International Conference on Management of data*, San Jose, California, 1995 (also appeared in an extended version as Technical Report CS-TR-3383, Dept. of Computer Science, University of Maryland, 1994)
-

- [Fen92] Fenton, Norman E. *Software Metrics – A Rigorous Approach*. Reader in Software Engineering, Chapman & Hall, London, 1992
- [FFR96] Farquhar, A., Fikes, R., Rice, J.: The Ontolingua Server: a Tool for Collaborative Ontology Construction. In: *Proc. Of the Knowledge Acquisition Workshop*, Banff, Canada, 1996
- [Fill68] Fillmore, C.: The Case for Case, In: Bach, E., Harms, R. (Hg.): *Universals in Linguistic Theory*, Holt, New York, 1968, S. 1-88
- [Fisc98] Fischer, Bernd. Specification-Based Browsing of Software Component Libraries. *Proc. of ASE-98: The 13th IEEE Conf. on Automated Software Engineering*, Honolulu, Hawaii, 1998
- [FKM00] Homepage Forschungskolleg Medien und kulturelle Kommunikation, <http://www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/FKMedien/>, 2000
- [FLGD87] Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11):964-971, 1987
- [FrCo91] Frakes, B., Cox, C. Implementation of Porter's stemming algorithm for English (1986–1991). Available at <http://www.cs.jhu.edu/~weiss/ir.html>
- [FrRe91] Fruchtermann, T., Reingold, E. Graph-drawing by force-directed placement. *Software Practice and Experience*, 21(11):1129-1164, 1991
- [FuGr00] Fuhr, Norbert, and Großjohann, Kai. XIRQL – An Extension of XQL for Information Retrieval. *ACM SIGIR 2000 Workshop On XML and Information Retrieval*, Athens, Greece, 2000
- [GaRo97] Gaizauskas, Robert, and Alexander M. Robertson. Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web. *Proceedings of RIAO 97: Computer-Assisted Information Searching on the Internet*, Montreal, Canada, 1997, pp. 356-370
- [GCO99] GLOBAL CAPE OPEN Web Page: <http://www.global-cape-open.org>, 1999
- [GeEi98] Gershon, Nahum, Eick, Stephen G. Guest Editors' Introduction: Information Visualization - The Next Frontier. In: *Journal of Intelligent Information Systems*, Vol. 11, No. 3, 1998, pp.199-204
- [Gilb93] Girardi, M.R., Ibrahim, B. An approach to improve the effectiveness of software retrieval. *Proceedings of the 3rd Irvine Software Symposium*, Irvine, CA, 1993
- [Green99] Green, Stephen J. Building Hypertext Links By Computing Semantic Similarity. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 5, September/October 1999, pp. 713–730
- [GWL+95] Gershon, N., Winstead, J., LeVasseur, J., Croall, J., Pernick, A., Ruh, W. Visualizing World Wide Web Information Resources. *Proc. of CIKM'95 Workshop on New Paradigms in Information Visualization and Manipulation*, Baltimore, Maryland, USA, 1995
- [HaBe92] Hargreaves, Kathryn, Berry, Karl: *Regex – GNU regular expression library manual*, <ftp.gnu.org>, 1992, GNU regular expression library, version 0.12, <ftp://ftp.gnu.org/gnu/regex/regex-0.12.tar.gz>, as of April 2000
- [HaRo00] Hahn, Udo, and Martin Romacker. Content Management in the SynDiKATe system – How technical documents are automatically transformed to text knowledge bases. *Data & Knowledge Engineering* 35, 2000, pp. 197–159
- [Haro98] Harold, Elliotte Rusty: *XML: Extensible Markup Language*, IDG Books, 1998

-
- [HDWB95] Hendley, R.J., Drew, N.S., Wood, A.M., Beale, R. Narcissus: Visualizing Information. Proc. of Information Visualization Symposium '95, IEEE Press, Atlanta, 1995, pp. 90–96
- [Hea95] Hearst, Marti A. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In: Proceedings of CHI 95, Denver, Colorado, 1995
- [Hea99] Hearst, Marti. User Interfaces and Visualization. In: Ricardo Baeza-Yates, Berthier Ribeiro-Neto (eds.): Modern Information Retrieval, Chapter 10. Addison-Wesley-Longman Publishing co., 1999
- [HeFi01] Heckert, Allan, and James F. Filliben. Exploratory Data Analysis. In: Carroll Croarkin and Paul Tobias (eds.): Engineering Statistics Handbook, Chapter 1, 2001, <http://www.itl.nist.gov/div898/handbook>
- [HeFr96] Hertzum, Morten and Eric Frokjaer. Browsing and Querying in Online Documentation: A Study of User Interfaces and the Interaction Process. ACM Transactions on Human Computer Interaction, 3(2):136–161, 1996
- [HeKa97] Hearst, M.A., Karadi, Ch. Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. In Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, Pennsylvania, USA, July, 1997
- [HeMi98] Hetzler, Beth, Miller, Nancy. Four Critical Elements for Designing Information Exploration Systems. In: Proc. of the Information Exploration Workshop for ACM SIGCHI '98, Los Angeles, 1998
- [HePe96] Hearst, Marti A., Pedersen, Jan O. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zürich, Switzerland, August, 1996
- [Her98] Heringer, Hans Jürgen. Das höchste der Gefühle – Empirische Studien zur distributiven Semantik. Verlag Stauffenburg, Tübingen, 1998
- [HGKP98] Himmel, D., Greaves, M. Kao, A., Poteet, S. Visualization for Large Collections of Multimedia Information. Workshop on Content Visualization and Intermedia Representations, University of Montreal, Montreal, Quebec, Canada, 1998
- [HHHW98] Hetzler, Beth, Harris, W. Michelle, Havre, Susan, Whitney, Paul. Visualizing the Full Spectrum of Document Relationships. In: Structures and Relations in Knowledge Organization. Proc. of the 5th International ISKO Conference, Würzburg, ERGON Verlag, 1998, pp. 168–175
- [HKLK96] Honkela Timo, Kaski Samuel, Lagus Krista, and Kohonen Teuvo. Newsgroup Exploration with WEBSOM Method and Browsing Interface. Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996
- [HKLK97] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. WEBSOM – self-organizing maps of document collections. In: Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, 1997, pp. 310–315
- [HoMa98] Hovy, Ed, and Daniel Marcu. Automated Text Summarization. Pre-conference tutorial at the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '98), Université de Montréal, Montréal, Québec, Canada, August 9, 1998. Tutorial notes are available at <http://www.isi.edu/~marcu/coling-acl98-tutorial.html>
- [Honk97] Honkela, Timo. Self-Organizing Maps in Natural Language Processing. PhD thesis, Helsinki University of Technology, Neural Networks Research Centre, Helsinki, 1997
-

- [HWM98] Hetzler, Beth, Whitney, Paul, Martucci, Lou, Thomas, Jim. Multi-facetted Insight through Interoperable Visual Information Analysis Paradigms. In: Proceedings of IEEE Symposium on Information Visualization, InfoVis '98, Research Triangle Park, North Carolina, 1998, pp. 137–144
- [ISO92] ISO 10241: International standard – International terminology standards – Preparation and layout (1st ed.). International Organization for Standardization, Geneva, 1992
- [Jaco92] Jacobson, Ivar; Christerson, Magnus; Jonsson, Patrik; Övergaard, G.: Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, Reading, 1992
- [Jäg01] Jäger, Ludwig. Transkriptivität – Zur medialen Logik der kulturellen Semantik. In: Ludwig Jäger, Georg Stanitzek (Hrsg.): Transkribieren – Medien/Lektüre, München, 2001
- [JaKe99] Jarke, M., Kethers, S.: Regionale Kooperationskompetenz: Probleme und Modellierungstechniken. *Wirtschaftsinformatik*, 41(4):316-325, 1999
- [JBK+99] Jarke, M.; Becks, A.; Köller, J.; Tresp, C.; Braunschweig, B.: Designing Standards for Open Simulation Environments in the Chemical Industries: A Computer-Supported Use-Case Approach. *Systems Engineering - Sharing the Future: Proceedings of the Ninth Annual International Symposium of the International Council on Systems Engineering*, Brighton, England, June, 1999
- [JFM97] Joachims, Th., Freitag, D., Mitchel, T. WebWatcher: A Tour Guide for the World Wide Web. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997
- [JoMI98] Joachims, Th., Mladenec, D. Browsing-Assistenten, Tour Guides und adaptive WWW-Server. *Künstliche Intelligenz*, Heft 3/98, Seiten 23-29, 1998
- [JTC98] Jarke, M.; Tung Bui, X.; Carroll, J. M. Scenario Management: An Interdisciplinary Approach. *Requirements Engineering* 3:155–173, 1998
- [KaKa89] Kamada, T., Kawai, S. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7-15, 1989
- [KeKr96] Keim, Daniel A., Hans-Peter Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, December 1996, pp. 923–938
- [KHLK98] Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. WEBSOM – self-organizing maps of document collections. *Neurocomputing*, volume 21, 1998, pp. 101-117
- [KKLH96] Kohonen, T., Kaski, S., Lagus, K., and Honkela, T. Very large two-level SOM for the browsing of newsgroups. In: von der Malsburg, C., von Seelen, W., Vorbrüggen, J. C., and Sendhoff, B., editors, *Proceedings of ICANN96 International Conference on Artificial Neural Networks*, Bochum, Germany, Lecture Notes in Computer Science, LNCS 1112), Springer, Berlin, 1996, pp. 269–274
- [KlFo88] Klir, George J. and Tina A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, 1988
- [KLHK98] Kaski, S., Lagus, K., Honkela, T., and Kohonen, T. Statistical aspects of the WEBSOM system in organizing document collections. *Computing Science and Statistics* 29, 1998, pp. 281-290
- [KlJa99] Klamma, R., Jarke, R.: Knowledge Management Cultures: A Comparison of Engineering and Cultural Science projects. *ECSCW-Workshop XMWS'99, Beyond Knowledge Management: Managing Expertise*, 13.9.99, Kopenhagen, Denmark, 1999

-
- [KLP96] Kleiboemer, Adrience J., Manette B. Lazear, and Jan O. Pedersen. Tailoring a Retrieval System for Naïve Users. Proc. of the 5th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA, 1996
 - [Koho82] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43 (1): 59 – 69, 1982
 - [Koho95] Kohonen, T. *Self-Organizing Maps*. Springer, Berlin, 2nd Edition, 1995
 - [Koho98] Kohonen, T. Self-organization of very large document collections: State of the art. In Niklasson, L., Bodén, M., and Ziemke, T., editors, *Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks*, volume 1, pages 65-74. Springer, London, 1998
 - [Korf97] Korfhage, Robert R. *Information Storage and Retrieval*. John Wiley and Sons, New York, 1997
 - [KPJ00] Klamma, R., Peters, P., Jarke, M.: Vernetztes Verbesserungsmanagement. *Wirtschaftsinformatik*, 42(1):15-26, 2000
 - [Lagus98] Lagus, K. Generalizability of the WEBSOM method to document collections of various types. In Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98), volume 1, pages 210-214, Aachen, Germany, 1998
 - [LaNa95] Langer, H., Naumann, S. Parsing natürlicher Sprache. In: Günther Görz (Hrsg.): *Einführung in die Künstliche Intelligenz*, Addison-Wesley, Bonn, 1995
 - [Larm98] Larman, Craig. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Prentice Hall, New Jersey, 1998
 - [LCDL00] Luk, Robert, Alvin Chan, Tharam Dillon, and H.V. Leong. A Survey of Search Engines for XML Documents. *ACM SIGIR 2000 Workshop On XML and Information Retrieval*, Athens, Greece, 2000
 - [LeGa94] Lewis, David D., Gale, William A. A Sequential Algorithm for Training Text Classifiers. In: *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 1994, pp. 3-12
 - [Leng96] Lengnink, Katja. *Formalisierung von Ähnlichkeit aus Sicht der Formalen Begriffsanalyse*. Dissertation, Fachbereich Mathematik, Technische Hochschule Darmstadt, 1996
 - [Lenz98a] Lenz, Mario. Managing the Knowledge Contained in Technical Documents. In U. Reimer (ed.): *Proc. of the 2nd International Conference on Practical Aspects of Knowledge Management*, Basel, Switzerland, 1998
 - [Lenz98b] Lenz, Mario. Textual CBR and Information Retrieval – A Comparison. In Gierl, Lenz (eds.): *Proceedings of the 6th German Workshop on Case-Based Reasoning*, IMIB Report, University of Rostock, 1998
 - [Lewis91] Lewis, David D. Natural Language Processing and Text Classification: Position Paper. *Proceedings of the Workshop on Future Directions in Text Analysis, Retrieval and Understanding*, Chicago, IL, 1991, pp. 52 – 57
 - [Lewis92] Lewis, David. Text Representation for Intelligent Text Retrieval: A Classification-Oriented View. In: Paul S. Jacobs (ed.): *Text-Based Intelligent Systems*, Chapter 9, Lawrence Erlbaum, 1992
 - [LHKK96] Lagus, Krista, Honkela, Timo, Kaski, Samuel, Kohonen, Teuvo. Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Menlo Park, CA, 1996, pp. 238-243
-

- [Lik32] Likert, Rensis. A technique for the measurement of attitudes. *Archives of Psychology*, 140, June, 1932
- [Lin92] Lin, X. Visualization for the document space. In: *Proceedings of Visualization 92*, Boston, October 21 – 23, 1992, pp 274-281
- [Lin95] Lin, Xia. Searching and Browsing on Map Displays. In: *Proceedings of the ASIS '95 Annual Conference*, American Society for Information Science, October 9 – 12, Chicago, 1995
- [Lin96] Lin, Xia. Graphical Table of Contents. In: *Proceedings of the 1st ACM International Conference on Digital Libraries*, March 20-23, 1996, Bethesda, Maryland. ACM Press, 1996
- [Lin97] Lin, Xia. Map Displays for Information Retrieval. *Journal of the American Society for Information Science*, 48:40-54, 1997
- [LoP197] López de Mántaras, Ramon and Enric Plaza. Case-Based Reasoning: An Overview. *AI Communications* 10:21-29, 1997
- [Lore84] Lorenz, Rolf J. *Grundbegriffe der Biometrie*. Gustav Fischer Verlag, Stuttgart, 1984
- [LSA+94] Lehnert, W., S. Soderland, D. Aronow, F. Feng, and A. Shmueli. Inductive Text Classification for Medical Applications. *Journal for Experimental and Theoretical Artificial Intelligence (JETAI)*. 7(1), pp. 49-80, 1994, also available as technical report TC-32 at the Center for Intelligent Information Retrieval, University of Massachusetts
- [LSM91] Lin, Xia, Soergel, Dagobert, Marchionini, Gary. A Self-Organizing Semantic Map for Information Retrieval. In: *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1991, pp. 262–269
- [Lyon81] John Lyons. *Language and Linguistics – An Introduction*. Cambridge University Press, Cambridge, 1981
- [Mar92] Marchionini, G. Interfaces for End-User Information Seeking. *Journal of the American Society for Information Science*, 43(2): 156–163, 1992
- [Math97] Mathar, R. *Multi-Dimensionale Skalierung*. Teubner Verlag, Stuttgart, 1997
- [MBK91] Maarek, Y.; Berry, D., Kaiser, G. An Information Retrieval Approach for Automatically Constructing Software Libraries. *IEEE Transactions on Software Engineering*, 17(8), pp. 800–813, 1991
- [Mead92] Meadow, Charles T. *Text information retrieval systems*. Academic Press, San Diego, California, 1992
- [Merk195a] Merkl, Dieter. A Connectionist View on Document Classification. *Proc. of the 6th Australian Database Conference (ADC'95)*, Adelaide, 1995, pp. 153–161
- [Merk195b] Merkl, Dieter. Content-Based Document Classification with Highly Compressed Input Data. In: *Proc. of the 5th International Conference on Artificial Neural Networks (ICANN'95)*, Paris, 1995, pp. 239–244
- [Merk195c] Merkl, Dieter. Content-Based Software Classification by Self-Organization. *Proc. of the IEEE Int'l Conference on Neural Networks (ICNN '95)*, Perth, Australia, 1995, pp. 1086–1091
- [Merk197a] Merkl, Dieter. Exploration of Document Collections with Self-Organizing Maps: A Novel Approach to Similarity Visualization. *European Symposium on Principles of Data Mining and Knowledge Discovery*, Trondheim, Norway, 1997

-
- [Merk197b] Merkl, Dieter. Exploration of Text Collections with Hierarchical Feature Maps. In: Proc. of the 20th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), Philadelphia, 1997
- [MHM98] Mourad Mechkour, David J. Harper, and Gheorghe Muresan. The WebCluster project: Using clustering for mediating access to the World Wide Web. Poster abstract, in: ACM-SIGIR'98, Melbourne, Australia, 1998
- [MHNW97] Miller, N.E., Beth Hetzler, Grant Nakamura, and Paul Whitney. The need for metrics in visual information analysis, in Workshop on New Paradigms in Information Visualization and Manipulation, Las Vegas, 1997
- [Miik90] Miikkulainen, R. Script recognition with hierarchical feature maps. *Connection Science*, 2, 1990
- [Mill95] Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39-41, 1995
- [Mins75] Minsky, M.: A Framework for Representing Knowledge. In: P.H. Winston (Hg.): *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, S. 211-277
- [Mit99] Mitchell, Tom. 20 Newsgroups Data. The UCI KDD Archive (<http://kdd.ics.uci.edu>), Information and Computer Science, University of California, Irvine, September 9, 1999,
- [MLKO98] Morse, E., M. Lewis, R. Korfhage, and K. Olsen. Evaluation of text, numeric and graphical presentations for information retrieval interfaces: User preference and task performance measures. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, 1998, pp. 1026-1031
- [MLO00] Morse, E., Lewis, M. and Olsen, K. Evaluating visualizations: Using a taxonomic guide. *International Journal of Human Computer Interaction*, 53(5), 2000, pp. 637-662
- [MMM98] Mili, A., Mili, R. and R.T. Mittermeir. A survey of software reuse libraries. *Annals of Software Engineering* 5, 1998, pp. 349-414
- [Mor99] Morse, Emile L. Evaluation of Visual Information Browsing Displays. PhD-Thesis, University of Pittsburgh, 1999
- [MRB89] Mendenhall W., Reimuth J. and Beaver R. *Statistics for Management and Economics* (6th edition), PWS-Kent, 1989
- [MTK94] Merkl, D., Tjoa, A Min, Kappel, G. A Self-Organizing Map that Learns the Semantic Similarity of Reusable Software Components. In: Proc. of the 5th Australian Conference on Neural Networks (ACNN '94), Brisbane, Australia, 1994, pp. 13-16
- [MuFo95] Mukherjea, Sougata, Foley, James D. Visualizing the World Wide Web with the Navigational View Builder. Technical Report 95-09, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 1995
- [MWBF98] Miller, N.E., Wong, P.C., Brewster, M., Foote, H. TOPIC ISLANDS – A Wavelet-Based Text Visualization System. In: Proc. of IEEE Visualization '98, 1998
- [NFH+96] Nowell, Lucy T., France, Robert K., Hix, Deborah, Heath, Lenwood, Fox, Edward A. Visualizing Search Results: Some Alternatives to Query-Document Similarity, In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, Switzerland, August, 1996
- [NJJ+96] Nissen, H.W., Jeusfeld, M.A., Jarke, M., Zemanek, G., Huber, H. Managing multiple requirements perspectives with meta models. *IEEE Software*, March 1996, pp. 37-48
-

- [NoTa97] Nonaka, Ikujiro, and Hirotaka Takeuchi. Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen. Campus Verlag, Frankfurt/Main, 1997
- [PNNL99] Pacific Northwest National Laboratory, Information Technology Showcase. Visual Text Analysis: SPIRE, 1999, <http://multimedia.pnl.gov:2080/showcase/>
- [PoHa95] Pohl, K.; Haumer, P. HYDRA: A Hypertext Model for Structuring Informal Requirements Representations. Proc. of the 2nd Int. Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ 95), Jyväskylä, Finland, 1995
- [Port80] Porter, M.F. An algorithm for suffix stripping. Program 14 (3), July 1980, pp. 130–137
- [PSHD96] Pirolli, Peter, Schank, Patricia, Hearst, Marti, Diehl, Christine. Scatter/gather browsing communicates the topic structure of a very large text collection. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Vancouver, 1996
- [Quil68] Quillian, M.R.: Semantic memory. In: Minsky, M. (Hg.): Semantic information processing, MIT Press, Cambridge, MA, 1968, S. 227-270
- [RBC+98] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N.A.M. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, P. Heymans. A Proposal for a Scenario Classification Framework. Requirements Engineering Journal, Vol. 3, No. 1, Springer Verlag, pp.23-47, 1998
- [Rieg83] Rieger, Burghard. Clusters in Semantic Space – Analysing natural language texts to model word meaning as a procedural representation. In: Delatte, L. (ed.): Actes du Congrès International Informatique et Science Humaines, Liège, Laboratoire d'Analyse Statistique des Langues Anciennes, 1983, pp. 805-814
- [Rieg85] Rieger, Burghard. Inducing a Relevance Relation in a Distance-like Data Structure of Fuzzy Word Meaning Representations. In: Allen, R.F. (eds.): Data Bases in the Humanities and Social Sciences, Proceedings of the 4th Int. Conference ICDBHSS/83, Osprey, FL, 1985, pp. 374-386
- [Rieg89] Rieger, Burghard. Unscharfe Semantik. Verlag Peter Lang, Frankfurt am Main, 1989
- [Rieg90] Rieger, Burghard. Unscharfe Semantik: zur numerischen Modellierung vager Bedeutungen von Wörtern als fuzzy Mengen. In: H.J. Friemel, G. Müller-Schönberg, A. Schütt (Hrsg.): Forum '90 – Wissenschaft und Technik. Neue Anwendungen mit Hilfe aktueller Computer Technologien, Informatik-Fachberichte 259, Springer, Berlin, 1990, S. 80–104
- [Rijs79] Rijsbergen, C.J. van. Information Retrieval. 2nd edition, Butterworths, London, 1979
- [RiKo89] Ritter, H., Kohonen, T. Self-organizing semantic maps. Biological Cybernetics 61, 1989, pp. 241–254
- [RiLe94] Riloff, E., and W. Lehnert. Information extraction as a basis for high-precision text classification. ACM Transactions on Information Systems, 12(3):296-333, 1994
- [Ritt91] Ritter, H. Asymptotic level density for a class of vector quantization process. IEEE Transactions on Neural Networks, Vol 2, 1991
- [RJB97] Rumbaugh, J.; Jacobsen, I. and Booch, G., Unified Modeling Language Reference Manual, Addison Wesley, 1997.
- [RMDT96] Risch, J., May, R., Dowson, S., Thomas, J. A Virtual Environment for Multimedia Intelligence Data Analysis. IEEE Computer Graphics and Applications, November 1996, pp. 33–41
- [RoCh98] Roussinov, D., Chen, H. A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation. Communication and Cognition – Artificial Intelligence, 15 (1–2), 1998, pp. 81–112

-
- [RoRa98] Roussinov, D., Ramsey, M. Information forage through adaptive visualization. In: Proceedings of the International ACM Conference on Digital Libraries, Pittsburgh, 1998
 - [Rous99a] Roussinov, Dimitri. Information Foraging Through Automatic Clustering and Summarization: A Self-Organizing Approach, Doctoral Dissertation, University of Arizona, August 1999
 - [Rous99b] Roussinov, Dimitri. Internet Search Using Adaptive Visualization. In: ACM SIGCHI '99 Conference on Human Factors in Computing Systems, Doctoral Consortium, Pittsburgh, May 1999
 - [SaBu88] Salton, G., Buckley, C. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24, 1988, pp 513–523
 - [Salt71] Salton, G. (Ed.): *The SMART Retrieval System – Experiments in Automatic Document Processing*. Englewood Cliffs, Prentice Hall, New Jersey, 1971
 - [Sand96] Sander, Georg. Visualisation of large graphs: Graph layout for applications in compiler construction. In *Proceedings of Order and Decision-Making*, Ottawa, Canada, 1996, available at <http://www.csi.uottawa.ca/ordal/>
 - [SBJ99] Sklorz, Stefan, Becks, Andreas, Jarke, Matthias. MIDAS – ein Multistrategiesystem zum explorativen Data Mining. *Data Mining und Data Warehousing als Grundlage moderner entscheidungsunterstützender Systeme (DMDW ,99)*, Magdeburg, September 1999, pp. 129–143
 - [SCGM00] Stasko J., R. Catrambone, M. Guzdial, K. McDonald. An Evaluation of Space-Filling Informational Visualizations for Depicting Hierarchical Structures. Technical Report GIT-GVU-00-03, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, 2000
 - [Schm94] Schmiedel, Albrecht. Semantic Indexing Based on Description Logics. In: *Reasoning about Structured Objects: Knowledge Representation Meets Databases*, Proceedings of 1st Workshop KRDB '94. Saarbrücken, Germany, 1994
 - [ScNa00] Schlieder, Torsten and Felix Naumann. Approximate Tree Embedding for Querying XML Data. *ACM SIGIR 2000 Workshop On XML and Information Retrieval*, Athens, Greece, 2000
 - [Seel01] Seeling, Christian. Eine vergleichende Studie zur Evaluierung der Aufgabenangemessenheit von Dokumentenlandkarten im Wissensmanagement. Diplomarbeit, RWTH Aachen, 2001
 - [SFRG99] Shneiderman B., D. Feldman, A. Rose, F.X. Grau. Visualizing Digital Library Search Results with Categorical and Hierarchical Axes. Technical Report CS-TR-3992, Department of Computer Science, University of Maryland, 1999
 - [Shan48] Shannon, Claude E. A Mathematical Theory of Communication. *Bell Systems Technical Journal* Vol. 27, No. 379–423, 1948, pp. 623–656
 - [Shn96] Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. Technical Report 96–66, Institute for Systems Research, University of Maryland, 1996
 - [Sklo01] Sklorz, Stefan. Exploratives Data Mining: Eine Integration numerischer und logischer sowie visueller und formaler Techniken des Data Mining. Dissertation, RWTH Aachen, 2001 (to appear)
 - [Sklo96] Sklorz, Stefan. A Method for Data Analysis Based on Self-Organizing Feature Maps. In: *Proceedings of the World Automation Congress*, Montpellier, France, 1996
 - [Small73] Small, H. Co-citation in the scientific literature: A new measure of the relationship between documents. *Journal of the American Society for Information Science*, 24, pp. 265–269
-

- [Sowa84] Sowa, S.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA, 1984
- [Stie93] Stieger, Daniel. Implementation of Porter's stemming algorithm for German (1993). Available at <http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat>
- [Stü95] Stürmer, Uta. Vergleichende Evaluierung verschiedener Interaktionsparadigmen für Information Retrieval. GMD-Studien Nr. 277, GMD-Forschungszentrum Informationstechnik GmbH, Sankt Augustin, 1995
- [SuEn96] Sutcliffe A. G., Ennis M. A Cognitive Model of Search Behaviour: a first step towards improving interface design for IR. Proceedings of BCS HCI Group Conference People and Computers XII, 1996
- [SVAH99] Simula, Olli, Vesanto, Juha, Alhoniemi, Esa, Hollmén, Jaakko. Analysis and Modeling of Complex Systems Using the Self-Organizing Map. In: N. Kasabov and R. Kozma. Neuro-Fuzzy Techniques for Intelligent Information Systems, Physica Verlag (Springer Verlag), 1999
- [SVM+99] Sebrechts, M., J. Vasilakis, M. S. Miller, J. V. Cugini, S. J. Laskowski. Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces. Proceedings of the 22nd Annual Int'l ACM-SIGIR Conf. on Research and Development in Information Retrieval, ACM Press, NewYork, 1999
- [SwAl96] Swan, Russel C., Allan, James. Improving Interactive Information Retrieval Effectiveness with 3-D Graphics, Tech. Rep. IR-100, Department of Computer Science, University of Massachusetts, Amherst, 1996
- [TBK+96] Tresp, C., Becks, A., Klinkenberg, R. Hiltner, J. Knowledge Representation in a World with Vague Concepts. Proceedings of Intelligent Systems: A Semiotic Perspective, pp. 71–76, Gaithersburg, Maryland, USA, 1996
- [TeMo97] Teufel, S., and M. Moens. Sentence Extraction as a Classification Task. Workshop on Intelligent and Scaleable Text Summarization, ACL/EACL 97, July 1997
- [Torg52] Torgerson, W.S. Multidimensional scaling: I. Theory and method. Psychometrika 17, 1952, pp. 401–419
- [TrTü98] Tresp, C., and U. Tüben. Medical terminology processing for a tutoring system. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA98), Monash Univ., Australia, 1998
- [TrWa96] Troina, G., Walker, N. Document Classification and Searching – A Neural Network Approach. In: ESA Bulletin No. 87, European Space Agency Publications Division (EPD), August 1996
- [Tulsa97] Tulsa, O.K. Electronic Statistics Textbook. StatSoft Inc., 1997, <http://www.statsoft.com/textbook/stathome.html>
- [Tusk00] Tusk, Carsten. Development and Implementation of a Semantic Analysis Component for Technical Text Collections. Diplomarbeit, RWTH Aachen, 2000
- [UISi90] Ultsch, A., Siemon, H. Kohonen's self-organizing feature maps for exploratory data analysis. In: Proceedings of INNC'90, International Neural Network Conference, Dordrecht, NL, Kluwer, 1990
- [Ult93] Ultsch, A. Self-organizing neural networks for visualization and classification. In: Opitz, O., Lausen, B., and Klar, R (ed.): Information and Classification, Springer-Verlag, Berlin, 1993
- [VeBe96] Veerasamy, A., and Nicholas J. Belkin. Evaluation of a Tool for Visualization of Information Retrieval Results. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zürich, Switzerland, 1996

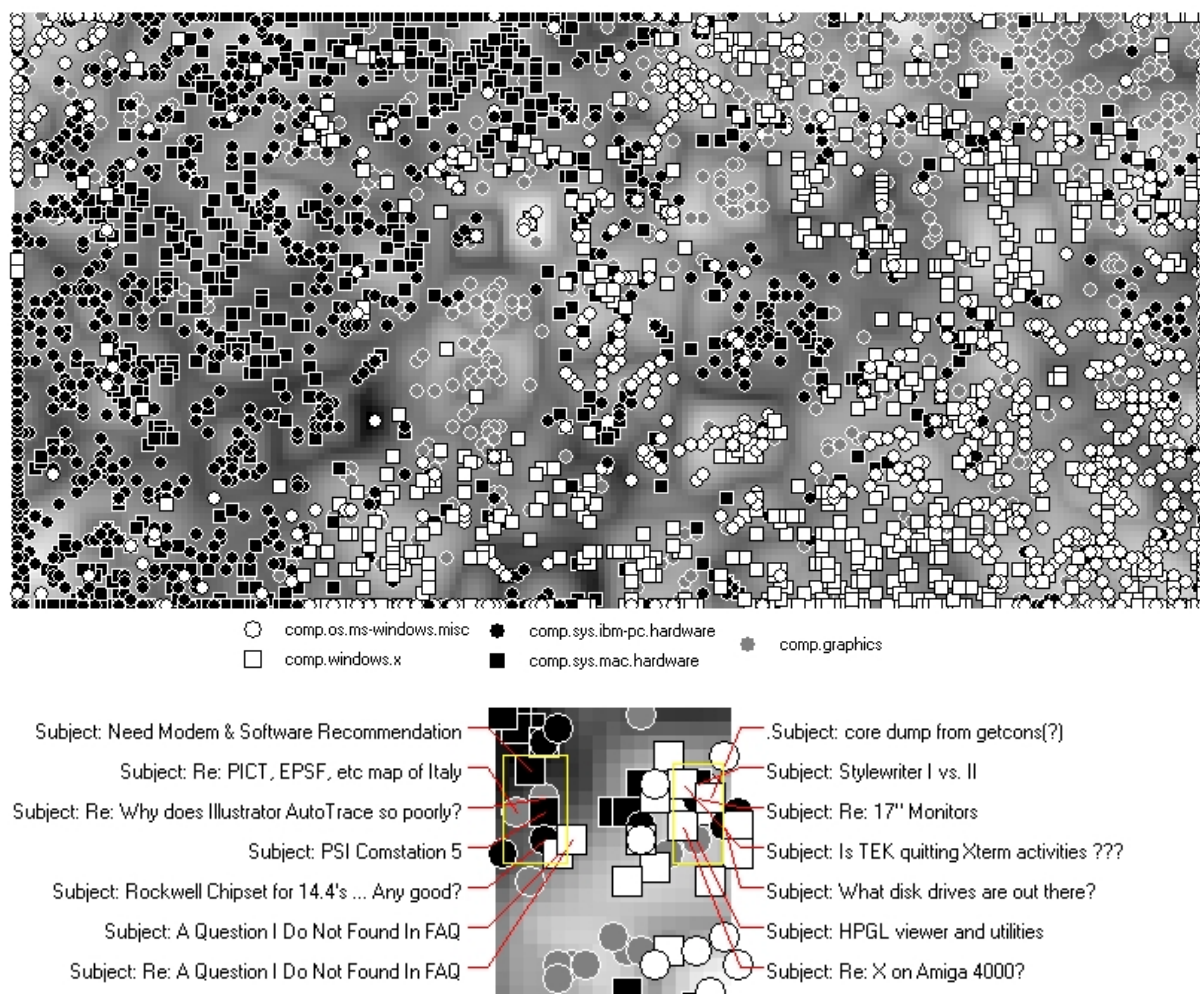
-
- [VeHe97] Veerasamy, A., Heikes, R. Effectiveness of a graphical display of retrieval results. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, Pennsylvania, USA, July, 1997
 - [Vesa99] Vesanto, Juha. SOM-Based Data Visualization Methods. Intelligent Data Analysis, Vol. 3, No. 2, Elsevier Science, April 1999
 - [Voor85] Voorhees, Ellen M. The Cluster Hypothesis Revisited. In: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, June, 1985, pp. 188-196
 - [VxHP99] VxInsight homepage: <http://www.cs.sandia.gov/projects/VxInsight.html>, see also: <http://www.quarks.de/spione/k06.htm>, 1999
 - [WaDa00] Warren, Paul and Graham Davies. Knowledge Management at BT Labs. Research, Technology, Management, Journal of the Industrial Research Institute, Washington DC, May-June, 2000
 - [WaRa98] Wang, W., Rada, R.. Structured Hypertext with Domain Semantics. In: ACM Transactions on Information Systems, Vol. 16, No. 4, October 1998, pp. 372-412
 - [WBJ90] Watzlawick, Paul, Janet H. Beavin, Don. D. Jackson. Menschliche Kommunikation. Formen, Störungen, Paradoxien. 8. Aufl., Bern 1990
 - [WeLe90] Wehrend, S., Lewis, C. A problem-oriented classification of visualization techniques. Proceedings of IEEE Visualization '90, pp. 139-143, IEEE Computer Society Press, 1990
 - [Will88] Willett, Peter. Recent Trends in Hierarchical Document Clustering: A Critical Review. Information Processing & Management, Vol. 24, No. 5, 1988, pp. 577 – 597
 - [WPJH98] Weidenhaupt, Klaus; Pohl, Klaus; Jarke, Matthias; Haumer, Peter: Scenarios in System Development: Current Practice. IEEE Software, March/April 1998
 - [WSB92] Wielinga, B., Schreiber, A.T. and Breuker, J.A. KADS: A Modeling Approach to Knowledge Engineering, in The KADS Approach to Knowledge Engineering Special Issue, Knowledge Acquisition, 4(1), Academic Press, London, UK, 1992
 - [WSHP99] WebSOM homepage: <http://websom.hut.fi/websom/>, 1999
 - [WSM96] Winiwarter, W. Schweighofer, E, Merkl, D. Knowledge Acquisition in Concept and Document Spaces by Using Self-organizing Neural Networks. In: S. Wermter, E. Riloff, G. Scheler (Eds.): Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, Lecture Notes in Artificial Intelligence 1040, Springer-Verlag, Heidelberg, 1996, pp. 75-86
 - [WTP+95] Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., Crow, V. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In: Proc. of IEEE Information Visualization 95 (InfoViz'95), 1995, pp. 51-58
 - [WuWi98] Wu, Mingfang, Wilkinson, Ross. Using Document Relationships for Better Answers. In: Proceedings of the Workshop on Principles of Digital Document Processing, St. Malo, France, 1998
 - [XBC94] Xu, Jinxi, Broglio, John, Croft, Bruce. The Design and Implementation of a Part of Speech Tagger for English. Technical Report 1994-060, University of Massachusetts at Amherst, 1994
 - [Yag80] Yager, R.R. On a general class of fuzzy connectives. Fuzzy Sets and Systems, 4, pp. 235-242, 1980
 - [Zad65] Zadeh, Lotfi. Fuzzy Sets. Information and Control 8, 1965, pp. 388ff
 - [Zad87] Zadeh, Lotfi. Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh (ed. by R.R. Yager et al.). John Wiley, New York, 1987
-

- [Zell94] Zell, Andreas. *Simulation Neuronaler Netze*. Addison-Wesley, Bonn, 1994
- [ZhFe98] Zhou, M.X., and S.K. Feiner. Visual Task Characterization for Automated Visual Discourse Synthesis. *Proc. ACM CHI '98*, April 18-23, Los Angeles, 1998., pp. 392–399

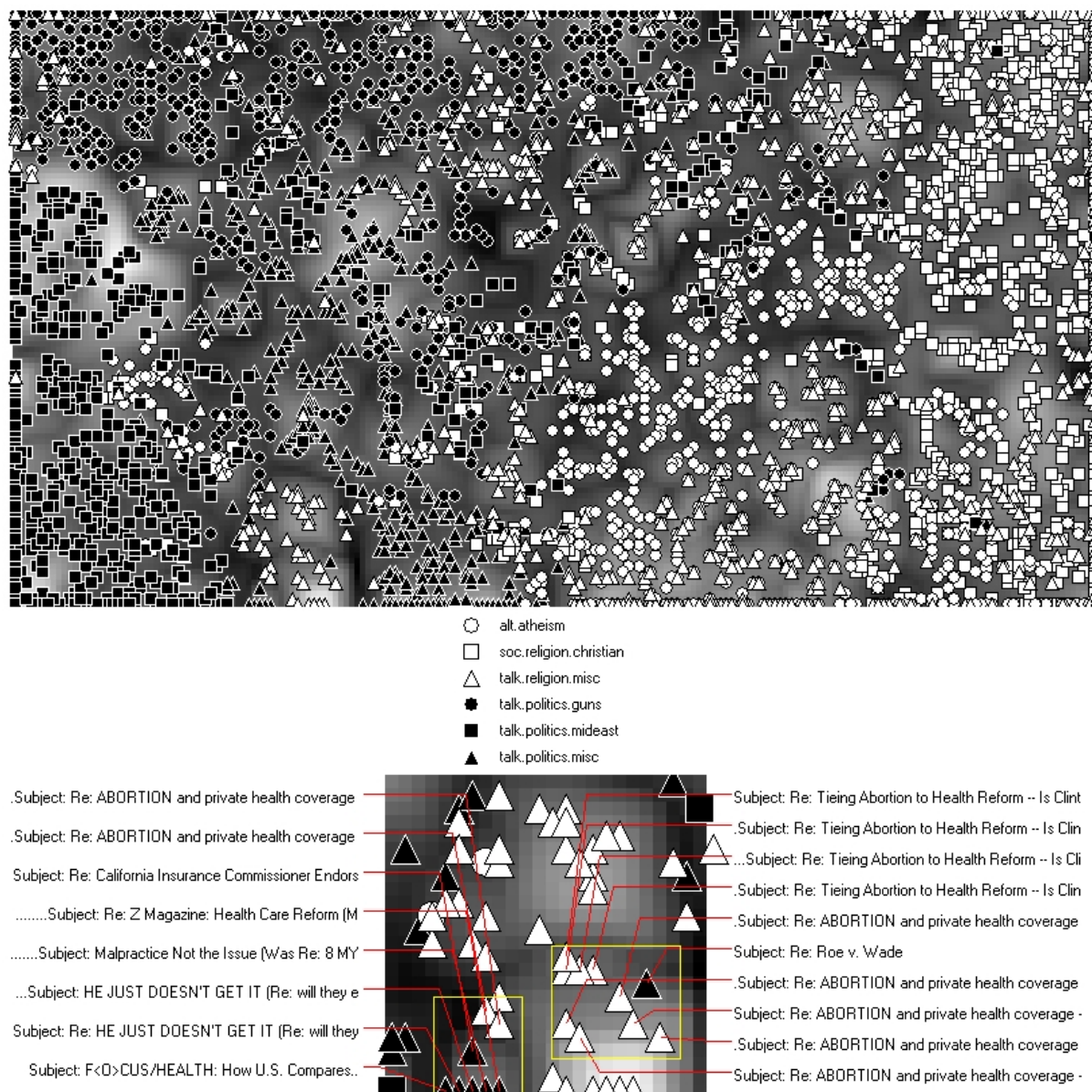
A Additional Experimental Newsgroup Document Maps

This appendix presents additional experimental document maps of a standardized test collection of newsgroups [Mit99]. For a detailed description of the collection and remarks on the interpretation of the results cf. section 6.5. The subset of newsgroups contained in each map is given in the symbol legends. These experiments with the basic document map technique aim at examining the scalability of the approach.

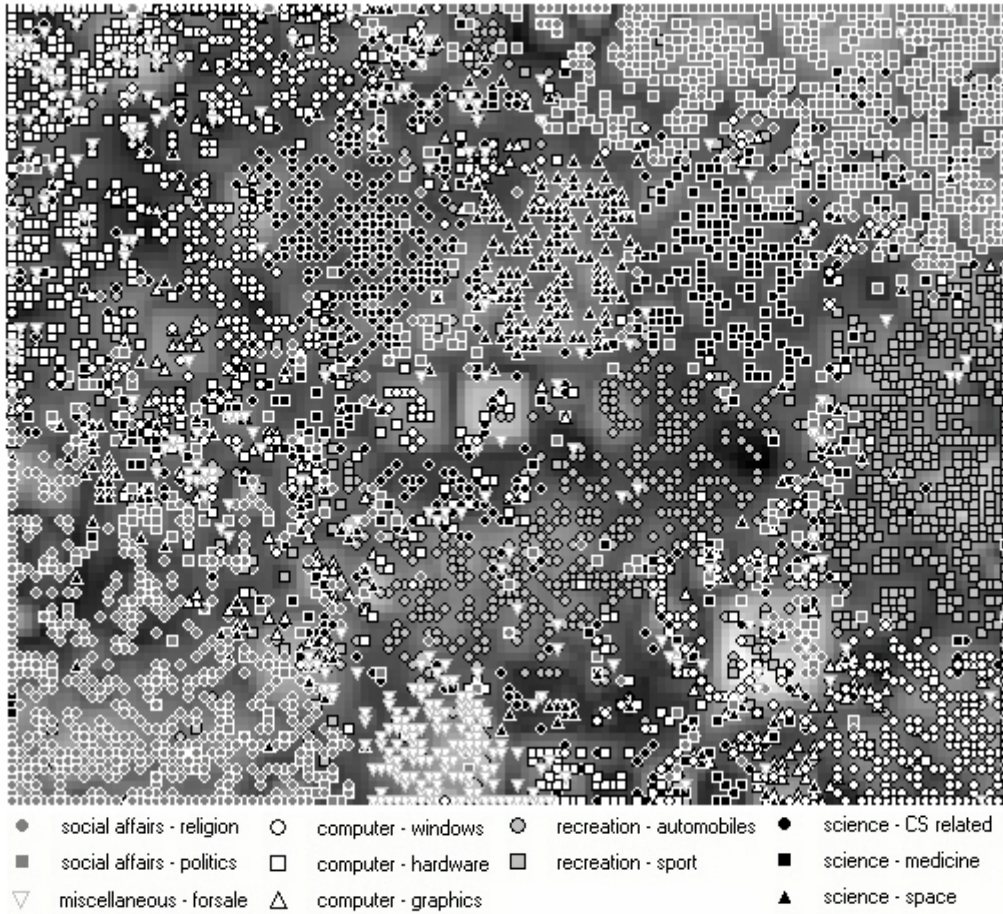
A.1 5,000 Newsgroup Articles Related to Computers



A.2 6,000 Newsgroup Articles Related to Social Affairs



A.3 10,000 Newsgroup Articles from 20 Newsgroups



Note: Cf. chapter 6.5 for newsgroups summarized in the categories above.

A.4 Parameters and Statistics of Maps and Collections

Note: All experiments have been performed on a Pentium II 350 MHz machine with 128 MB RAM and WinNT 4.0.

collection	name	newsgroups computer	20 newsgroups	newsgroups social affairs
	# documents	5,000	10,000	6,000
	avg. # words	201	244	352
indexing	size of vocabulary	37,429	55,203	31,974
	avg. # indexing terms / document	55	68	91
	weighting scheme	TF	TF	TF
	similarity measure	cosine	cosine	cosine
	dimension document space	100	200	150
SOM	size of SOM	200×100	125×100	200×110
	training steps per document	20	15	20
	basic map generation	9 h 45 min	17 h 19 min	18 h 36 min

B Original Task Sheets and Questionnaires for Comparative Study

B.1 Task Sheet T1

Aufgabe 1: Mit dem Inhalt der Textdatenbank vertraut machen

In dieser Aufgabe sollen Sie sich mit dem Inhalt der Textdatenbank vertraut machen und die thematische Struktur der Textsammlung untersuchen. Unter einem Thema der Textsammlung verstehen wir einen übergeordneten Sachgegenstand, der in mehreren (d.h. mindestens 2) Texten behandelt wird. Sie sollen sich nicht mit Einzeltexten beschäftigen, sondern sich rasch einen Überblick über alle Texte verschaffen. Sie haben für die Lösung der Aufgabe nur sehr begrenzte Zeit zur Verfügung.

Ihre Aufgabe:

- Untersuchen Sie zu welchen Themen oder Themenbereichen Texte vorhanden sind. Wenn Sie relativ große Themenbereiche finden, so versuchen Sie, Unterthemen zu erkennen.
- Machen Sie sich ein Bild über die ungefähre Anzahl von Texten zu den einzelnen Themen.

Nach der Bearbeitungszeit werden Ihnen Themen vorgeschlagen, und Sie sollen die Themen ankreuzen, zu denen Texte vorhanden sind. Für jedes vorhandene Thema sollen Sie eine möglichst genaue Schätzung für die Anzahl der Texte angeben, die dazu gehören. Sie bekommen später einen Bearbeitungsbogen, der folgendermaßen aufgebaut ist:

Thema	Texte gefunden?		Anzahl Texte	
	ja	nein	Anz.	weiß nicht
Fernsehen	ja	nein	#	?
....	ja	nein	#	?

Machen Sie sich Notizen, um nach der Bearbeitungszeit die erforderlichen Angaben machen zu können!

Beachten Sie:

- Bei der Bewertung Ihrer Lösung kommt es auf Genauigkeit und auf Vollständigkeit an.
- Nutzen Sie eventuell überschüssige Zeit, um die Qualität Ihrer Lösung zu verbessern.
- Die Titel der Texte beschreiben den Textinhalt nicht immer ausreichend. Untersuchen Sie im Zweifel solche Texte genauer.

Sie haben zur Bearbeitung 8 Minuten Zeit.

B.2 Answer Sheet T1 (Front)

Lösungszettel zu Aufgabe 1 (Mit dem Inhalt der Textdatenbank vertraut machen)

Auf der Rückseite dieses Zettels finden Sie eine wie folgt aufgebaute Tabelle:

Thema	Texte gefunden?		Anzahl Texte	
	ja	nein	Anz.	weiß nicht
Fernsehen	ja	nein	#	?
....	ja	nein	#	?

Kreuzen Sie Themen an, zu denen Texte vorhanden sind. Geben Sie für vorhandene Themen eine möglichst genaue Schätzung für die Anzahl der Texte an, die dazu gehören. Bitte vermeiden Sie Angaben der Art „5-10 Texte“. Gehen Sie beim Ausfüllen der Tabelle wie folgt vor:

1. Sie meinen, daß zum Thema „Fernsehen“ ungefähr 7 Texte vorhanden sind:

Fernsehen	<input checked="" type="checkbox"/>	nein	7	?
-----------	-------------------------------------	------	---	---

2. Sie meinen, daß zum Thema „Fernsehen“ Texte vorhanden sind, können aber die ungefähre Anzahl nicht schätzen:

Fernsehen	<input checked="" type="checkbox"/>	nein	#	<input checked="" type="checkbox"/>
-----------	-------------------------------------	------	---	-------------------------------------

3. Sie meinen, daß zum Thema „Fernsehen“ keine Texte vorhanden sind:

Fernsehen	ja	<input checked="" type="checkbox"/>	#	?
-----------	----	-------------------------------------	---	---

Tragen Sie nun bitte Ihre Lösung auf der Rückseite ein!

B.3 Answer Sheet T1 (Back)

Thema	Texte gefunden?		Anzahl Texte		Thema	Texte gefunden?		Anzahl Texte	
	ja	nein	Anz.	weiß nicht		ja	nein	Anz.	weiß nicht
Hera Lind	ja	nein	#	?	Elektronische Bücher	ja	nein	#	?
Stephen King	ja	nein	#	?	Handhelds / PDA's	ja	nein	#	?
Harry Potter	ja	nein	#	?	Virtuelle Realität	ja	nein	#	?
Art Cologne	ja	nein	#	?	Automobilausstellung IAA	ja	nein	#	?
Literarisches Quartett	ja	nein	#	?	ICE	ja	nein	#	?
Kunstgeschichte	ja	nein	#	?	Transrapid	ja	nein	#	?
Nobelpreis	ja	nein	#	?	Concorde-Absturz	ja	nein	#	?
Forschungspolitik	ja	nein	#	?	Russisches Atom-U-Boot	ja	nein	#	?
Hirnforschung	ja	nein	#	?	Wirtschaftsgipfel	ja	nein	#	?
Krebsforschung	ja	nein	#	?	Einführung des Euro	ja	nein	#	?
BSE / Rinderwahn	ja	nein	#	?	Atomkraftwerke	ja	nein	#	?
Biologische Viren	ja	nein	#	?	Kartellverfahren Microsoft	ja	nein	#	?
Meeresbiologie	ja	nein	#	?	Städteplanung	ja	nein	#	?
Sexualität	ja	nein	#	?	Reichstagsumbau Berlin	ja	nein	#	?
Menschliche Vorfahren	ja	nein	#	?	Oktoberfest	ja	nein	#	?
Mammuts / Dinosaurier	ja	nein	#	?	Präsidentschaftswahl USA	ja	nein	#	?
Erneuerbare Energie	ja	nein	#	?	Raumsonden / Satelliten	ja	nein	#	?
Ozonloch	ja	nein	#	?	Weltraumteleskop Hubble	ja	nein	#	?
Intelligente Häuser	ja	nein	#	?	Raumfahren	ja	nein	#	?
Siamesische Zwillinge	ja	nein	#	?	Schwarze Löcher	ja	nein	#	?
Klonen	ja	nein	#	?	Sternzeichen	ja	nein	#	?
Militärtechnologie	ja	nein	#	?	Asteroiden & Meteoriten	ja	nein	#	?
Ägyptische Archäologie	ja	nein	#	?	Sonnenfinsternis / Sonne	ja	nein	#	?
Computerviren	ja	nein	#	?	Historie der Mondlandung	ja	nein	#	?
Softwarepatente	ja	nein	#	?	Weltraumstation	ja	nein	#	?
Internet-Portale	ja	nein	#	?	Überschwemmungen	ja	nein	#	?
Handy / UMTS	ja	nein	#	?	Erdbebenforschung	ja	nein	#	?
e-Commerce	ja	nein	#	?	Vorhersage von Hurrikans	ja	nein	#	?
Roboter / RoboCup	ja	nein	#	?	Vulkanausbrüche	ja	nein	#	?

B.6 Task and Answer Sheet T4

Aufgabe 4: Textgruppen nach inhaltlichen Kriterien bilden

In dieser Aufgabe sollen Sie die inhaltliche Zusammengehörigkeit von Texten untersuchen. Einige Texte der Sammlung sind blau eingefärbt. Beschäftigen Sie sich nur mit diesen eingefärbten Texten. Finden Sie heraus, welche Texte inhaltlich möglichst gut zusammenpassen und bilden Sie Gruppen solcher inhaltlich verwandter Texte. Eine Gruppe darf zwischen 2 und 5 Texte umfassen. Eine optimale Lösung enthält 11 Textgruppen. Versuchen Sie, in der zur Verfügung stehenden Zeit möglichst viele und möglichst genaue Gruppen zu bilden.

Ihre Aufgabe:

- Finden Sie kleine Gruppen von Texten (2 bis 5 Texte), die inhaltlich stark verwandt sind. Jeder Text darf nur in genau einer Gruppe auftauchen. Notieren Sie die Nummern der Texte einer Gruppe in der Liste unten auf dieser Seite.
- Verschaffen Sie sich einen Überblick über den Inhalt der gefundenen Textgruppen (z.B. mit der Funktion „wichtige Sätze anzeigen“) und machen Sie sich Notizen. Im Anschluß an die Bearbeitungszeit werden Ihnen Aussagen präsentiert, und Sie sollen entscheiden, ob die Aussage eine mögliche Gruppe von Texten aus der blau markierten Menge charakterisiert.

Beachten Sie:

- Es kommt sowohl auf Genauigkeit als auch auf Vollständigkeit der Lösung an.
- Nutzen Sie überschüssige Zeit, um die Qualität Ihrer Lösung zu verbessern.
- Die Titel beschreiben den Textinhalt nicht immer ausreichend. Untersuchen Sie im Zweifel solche Texte genauer.

Sie haben zur Bearbeitung 10 Minuten Zeit.

Schreiben Sie bitte in die linke Spalte die Nummern der Texte, die Sie in einer Gruppe zusammenfassen würden. In der rechten Spalte können Sie sich Notizen zum Inhalt machen.

	Nummern der Dokumente	Ihre Notizen
Gruppe 1		
Gruppe 2		
Gruppe 3		
Gruppe 4		
Gruppe 5		
Gruppe 6		
Gruppe 7		
Gruppe 8		
Gruppe 9		
Gruppe 10		
Gruppe 11		

B.7 Additional Answer Sheet T4

Lösungszettel zu Aufgabe 4 (Textgruppen nach inhaltlichen Kriterien bilden)

Bitte kreuzen Sie in der Liste für jede Aussage an, ob diese eine mögliche Gruppe von Texten aus der blau markierten Menge charakterisiert, d.h. ob mindestens 2 Texte dieses Thema behandeln.

Aussage	Die Aussage charakterisiert eine Gruppe von Texten	
	ja	nein
Der Transport von Kommunikationssatelliten durch die Ariane-Trägerrakete soll billiger werden, um gegenüber amerikanischen Raumfahrtprojekten konkurrenzfähig zu bleiben.	ja	nein
Die Ariane-Rakete hat für verschiedene Länder Kommunikationssatelliten ins All gebracht	ja	nein
Zwei europäische Satelliten wurden ins All gebracht und haben die Aufgabe, die Magnetosphäre der Erde und die Sonnenwinde zu erforschen	ja	nein
Die Satelliten „Champ“ und „Cluster“ sind erfolgreich gestartet und haben Bilder von der Erdatmosphäre zur Erde gesendet.	ja	nein
Mit dem Weltraumteleskop „Hubble“ lassen sich Asteroiden, die sich auf Kollisionskurs mit der Erde befinden und eine Gefahr darstellen, früher finden als bisher.	ja	nein
Astronomen befürchteten, daß ein Asteroid auf der Erde einschlagen könnte, doch es wurde Entwarnung gegeben.	ja	nein
Die Raumsonde „Near“ umkreist die Erde und ist am Valentinstag in einer guten Position, um Bilder vom Asteroiden Eros zur Erde zu schicken.	ja	nein
Es wurde eine Raumsonde ins All geschickt, die zum ersten Mal einen Asteroiden umkreisen und erforschen soll.	ja	nein
Britische Forscher untersuchen die Einschlaggefahr von Himmelskörpern auf der Erde und untersuchen, welches Ausmaß eine Katastrophe haben könnte.	ja	nein
Es sollen Schutzmaßnahmen unternommen werden um die Chancen zu verringern, daß Asteroiden in Erdnähe kollidieren.	ja	nein
Um die Verschrottung der Mir zu verhindern, soll die Mir an der Börse gehandelt werden.	ja	nein
Geld der russischen Investorengruppe „MirCorp“ und verschiedener Millionäre wird dazu verwendet, die Raumstation Mir weiter zu betreiben.	ja	nein
Die Crew der Raumfähre Atlantis richtet die internationale Raumstation ISS mit möbliertem Wohnmodulen ein, damit Astronauten die Station beziehen können.	ja	nein

(Auf der Rückseite geht es weiter!)

Aussage	Die Aussage charakterisiert eine Gruppe von Texten	
	ja	nein
Die Raumfähre Atlantis hat am Wohnmodul der Raumstation ISS andockt und die Station bezogen. Die Atlantis soll ein defektes Wohnmodul von der Raumstation zur Erde zurückbringen.	ja	nein
Ein vergessenes Metallstück verhinderte den Start der Raumfähre Discovery zum 100. Jubiläumsflug.	ja	nein
Der 100. Jubiläumsflug der Discovery wurde wieder verschoben, weil beim Start ein Triebwerk versagte.	ja	nein
Es wurde ein neues Peilgerät entwickelt, mit dem die Welt von Satelliten aus genau vermessen werden kann.	ja	nein
Eine in Cape Canaveral gestartete Raumfähre mit deutscher Besatzung erstellt mit Radar und Sensoren eine genaue Karte der Welt.	ja	nein
Nach neuesten Untersuchungen haben personelle Mängel in der Entwicklungsabteilung der NASA einen technischen Defekt verursacht, der das Verschwinden der Marssonde Polar Lander zur Folge hatte.	ja	nein
Alle Versuche, wieder Kontakt zur verlorenen Marssonde Polar Lander herzustellen, schlugen fehl, und es wird nach den Ursachen für den Fehlschlag gesucht.	ja	nein
Verschiedene Sonden haben Bilder von der Oberfläche des Jupiters und der Jupiter-Mondes gemacht.	ja	nein
Die Marssonde Polar Lander ist außer Kontrolle geraten und im Kamikaze-Flug in die Marsatmosphäre eingetreten.	ja	nein

B.8 Questionnaires Task 1 – 4

Bitte kreuzen Sie für jede Frage die Stelle auf der Skala an, die Ihrer Meinung nach am besten zutrifft.

1) Wie gut war Ihrer Meinung nach das System geeignet, um die gestellte Aufgabe zu unterstützen?

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sehr schlecht	eher schlecht	mittelmäßig	eher gut	sehr gut

2) Wie gut schätzen Sie die Qualität Ihrer Ergebnisse ein?

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sehr schlecht	eher schlecht	mittelmäßig	eher gut	sehr gut

3) Wie anstrengend war die Lösung der Aufgabe mit dem System für Sie?

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sehr anstrengend	eher anstrengend	mäßig anstrengend	eher leicht	sehr leicht

B.9 Final Questionnaire

Abschließende Fragen

- 1) Sie haben 2 Systeme kennengelernt, mit denen sich Textdatenbanken inhaltlich erschließen lassen. Welches der beiden Systeme würden Sie vorziehen, wenn Sie ähnliche Aufgaben wie die hier behandelten in der Praxis durchführen müßten (Zutreffendes bitte ankreuzen)?
- ☐ Ich würde in jedem Fall das Dokumentenlandkartensystem einsetzen.
- ☐ Ich würde in jedem Fall das Textlistensystem einsetzen.
- ☐ Ich würde je nach Aufgabe das eine oder das andere System vorziehen. (Bitte machen Sie in der Tabelle je ein Kreuz für das System, das Sie zur Bearbeitung der jeweiligen Aufgabe lieber benutzen würden):

	Dokumentenlandkarte	Textliste
Mit dem Inhalt der Textdatenbank vertraut machen		
Thematische Ausreißer finden		
Eine Kategorisierung prüfen		
Textgruppen nach inhaltlichen Kriterien bilden		

- 2) Wie gut hat Ihnen die Arbeit mit dem Dokumentenlandkartensystem gefallen?

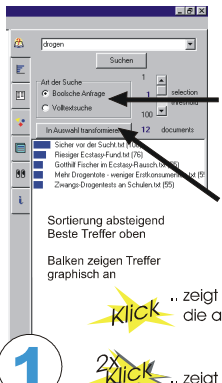
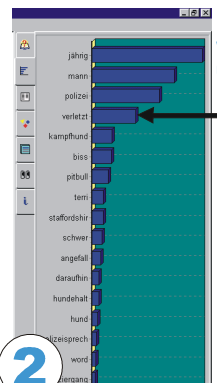
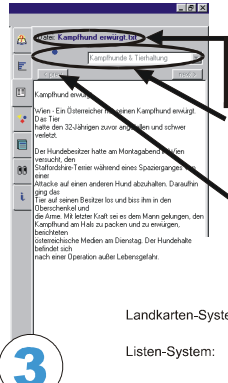

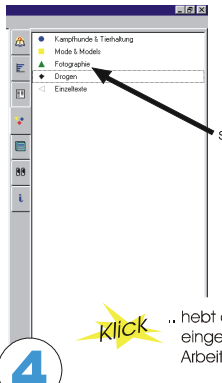
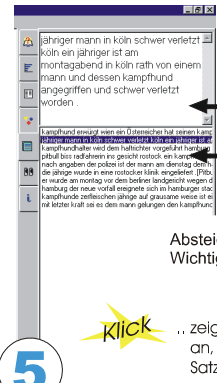

☐ ☐ ☐ ☐ ☐
sehr schlecht eher schlecht mittelmäßig eher gut sehr gut

- 3) Wie gut hat Ihnen die Arbeit mit dem Textlistensystem gefallen?

☐ ☐ ☐ ☐ ☐
sehr schlecht eher schlecht mittelmäßig eher gut sehr gut

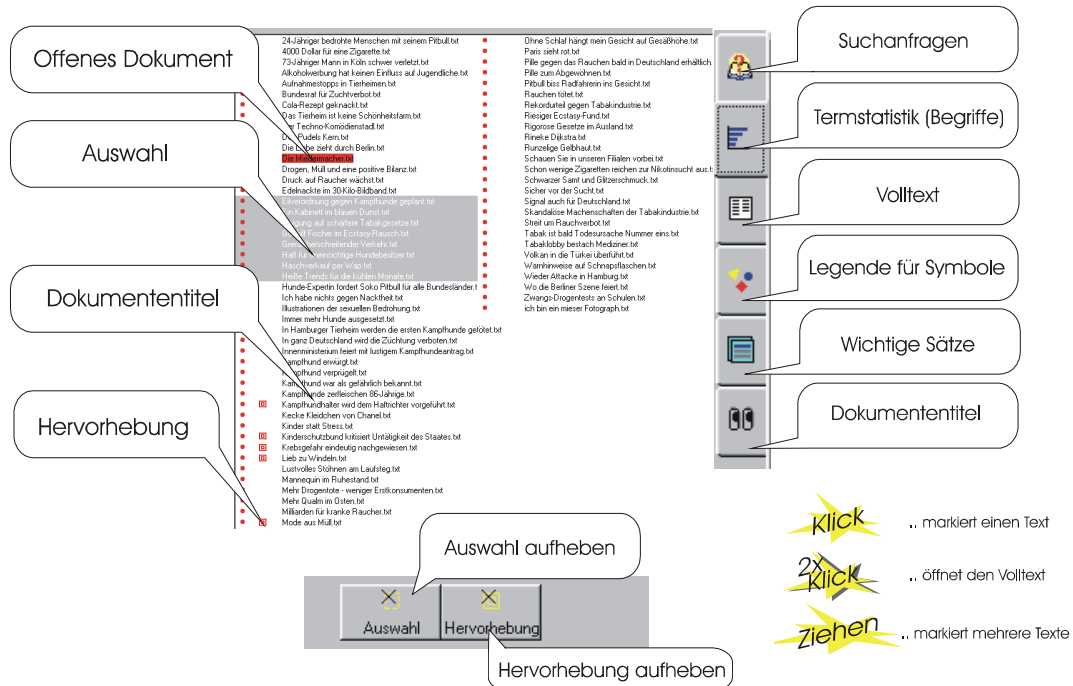
C GUI Quick Reference Sheets Used in Comparative Study

C.1 Functions of Both Systems

<p>Anfrage Suchanfrage in der Textsammlung</p>  <p>Boolesch: Mit Verknüpfungen: AND, OR, NOT, Klammern. Bsp.: drogen and (rauch or nadel)</p> <p>Volltext: Genau wie im Text. Bsp.: drogenkons</p> <p>Sortierung absteigend Beste Treffer oben</p> <p>Balken zeigen Treffer graphisch an</p> <p>1 klick .. zeigt auf der Arbeitsfläche die ausgewählten Texte an</p> <p>2 2x klick .. zeigt den Volltext an.</p>	<p>Begriffe Wichtige Schlagworte markierter Texte</p>  <p>Balken zeigen Wichtigkeit eines Begriffs grafisch an:</p> <p>Größe der Balken entspricht Signifikanz der Begriffe</p> <p>2</p>	<p>Text Volltext des geöffneten Textes</p>  <p>Titel Symbol & Klassenname</p> <p>Nur in Landkarten-System: Wechseln zwischen übereinanderliegenden Texten</p> <p>Landkarten-System: </p> <p>Listen-System: Titel.txt</p> <p>3</p>
<p>Legende Bedeutung der Textsymbole</p>  <p>Symbol und Beschreibung</p> <p>4 klick .. hebt die mit dem Symbol eingefärbten Texte in der Arbeitsfläche hervor.</p>	<p>Sätze Wichtige Sätze markierter Texte</p>  <p>Gut lesbare Version des unten markierten Satzes</p> <p>Wichtige Sätze der Texte, die in der Arbeitsfläche markiert sind</p> <p>Absteigend nach Wichtigkeit der Sätze sortiert</p> <p>5 klick .. zeigt auf der Arbeitsfläche an, zu welchem Text der Satz gehört.</p>	<p>Titel Titel markierter Texte</p>  <p>Absteigend alphabetisch sortiert</p> <p>6 klick .. zeigt auf der Arbeitsfläche an, zu welchem Text der Titel gehört.</p> <p>7 2x klick .. zeigt den Volltext an.</p>

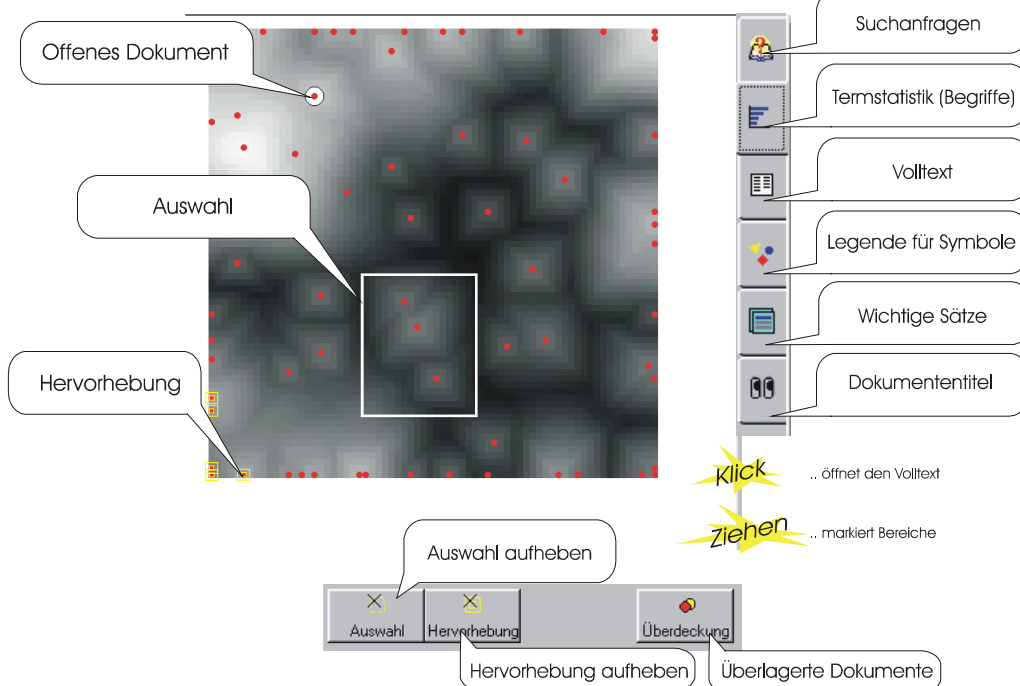
C.2 Workspace Document List

Arbeitsfläche Listensystem



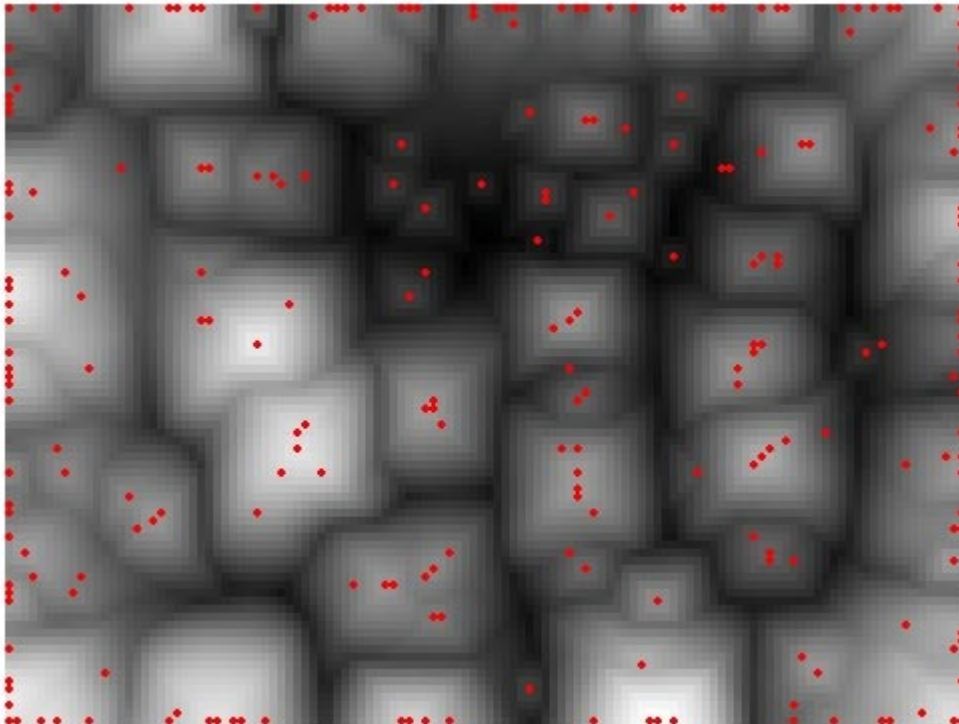
C.3 Workspace Document Map

Arbeitsfläche Landkartensystem



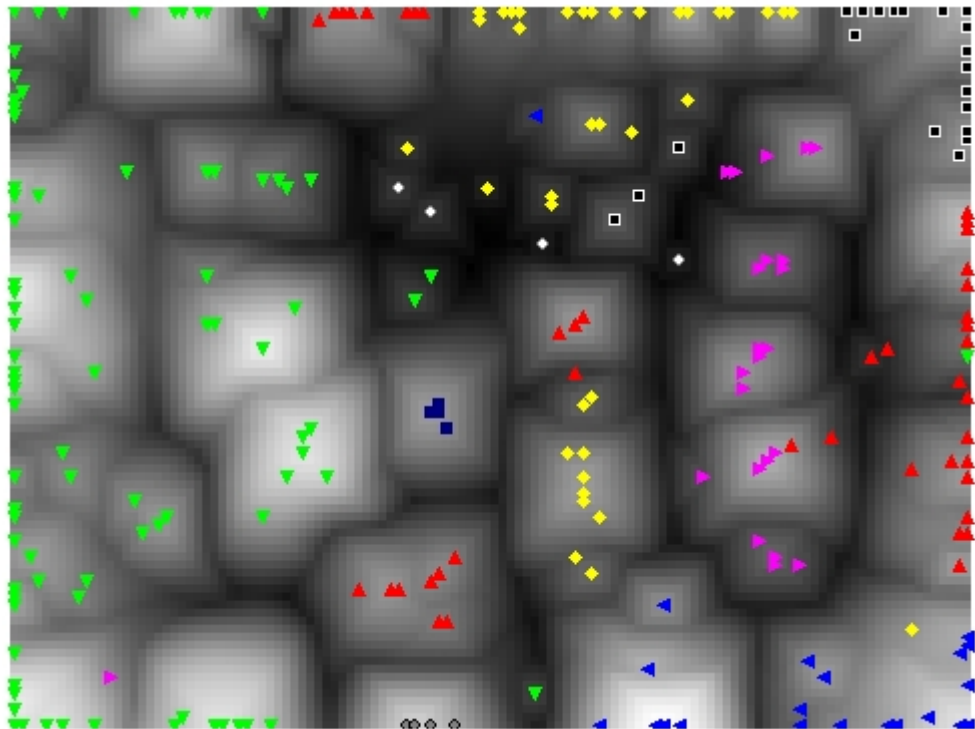
D Document Map Used in Comparative Laboratory Study

D.1 Basic Document Map



# documents	300
weighting scheme	TF
similarity measure	cos
dimension	150
avg. stress	0.088
size of SOM	120 × 90
training steps per document	30

D.2 Document Map and Part of Text List with Categories



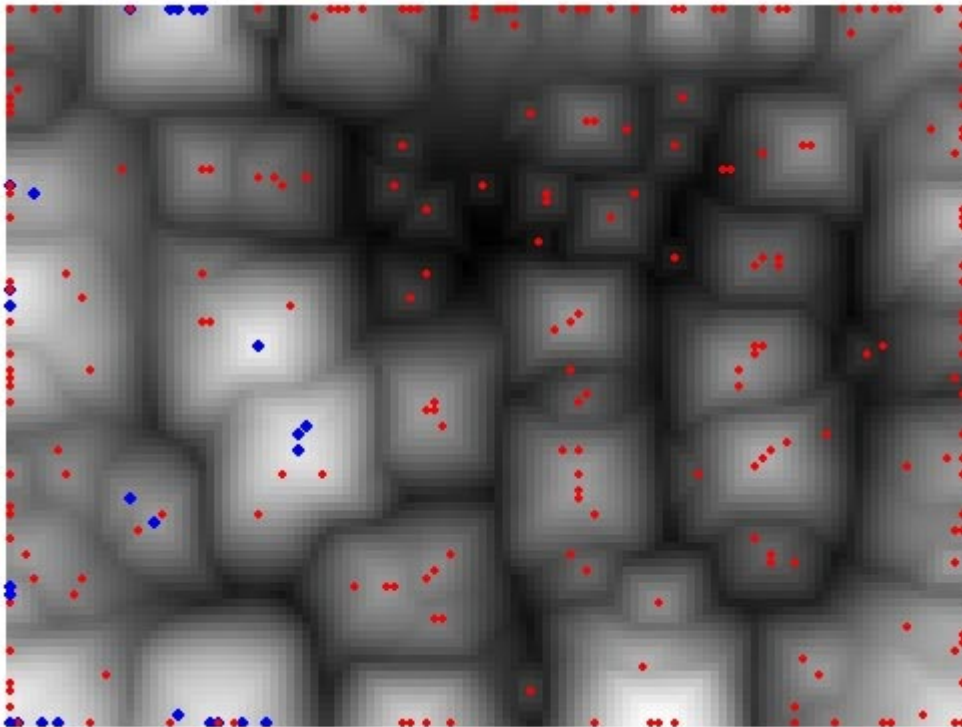
033-Harry Potter im Legoland.txt	065-Neuer BSE-Skandal.txt
034-Gericht stoppt illegale Potter-Übersetzungen.txt	066-Schreckenskönig oder Schreckgespenst.txt
035-US-Jäger vor gefährlicher Hirnerkrankung bei V	067-Kosmisches Schlüsselloch.txt
036-VIAGRA.txt	068-Naturschauspiel verwirrt Himmelsgucker.txt
037-Gehirnentwicklung erfolgt noch bis zur Pubertät	069-Cluster-Sonden gehen in Position.txt
038-Begleiter wird aus Klinik entlassen.txt	070-Jubeln, Jammern, Schlafen.txt
039-Wer baut die UMTS-Handys von morgen.txt	071-Die Handhelds werden attraktive Ziele für Sch
040-Mehdorn ist der Totengräber dieser Technolog	072-Die Jugend war noch nie so frei und ungezwun
041-Stephen King auf neuen Wegen.txt	073-Carlsen macht Kompromisse.txt
042-Begeisterung, Staus und erste Augenbeschwer	074-Die Kontaktfreudige.txt
043-Schafft die Esa was der Nasa nicht gelingt.txt	075-Die Mir soll an die Börse.txt
044-Asteroid verfehlt Erde nur knapp.txt	076-Neurobiologe und Kunstsammler.txt
045-Die Übertragung ist nur über Mücken möglich.t	077-Die schöne neue Welt nimmt Formen an.txt
046-Sony steigt in den Handheld-Markt ein.txt	078-Sonnenstürme bedrohlicher als erwartet.txt
047-Mars - Erde 20.txt	079-Hubble entdeckt verführerischen Schleier.txt
048-Patient starb an Lassa-Fieber.txt	080-Neue Besatzung zur Mir unterwegs.txt
049-Der Vater des Mikrochips.txt	081-Wunderschön und geräumig.txt
050-Forscher entdecken Zentrum des Sextriebs.txt	082-Umstrittene Sonde Cassini passiert die Erde.txt
051-Atlantis-Crew verlässt Raumstation.txt	083-US-Experten warnen vor Epidemie.txt
052-Karasek knallhart.txt	084-Dinosaurier-Ei samt Embryo gestohlen.txt
053-Blutanalyse im Raumanzug.txt	085-Meister des Knochenpuzzles.txt
054-Chris Columbus verfilmt Harry Potter.txt	086-Die WHO gibt teilweise Entwarnung.txt
055-Knapp 600 Millionen telefonieren per Handy.txt	087-Flugmodul 5,6,7 und 8 suchen noch einen Nar
056-Der erste Shopping-Assistent der ein Roboter is	088-Ebola-Virus.txt
057-Wegbereiter einer neuen chinesischen Literatur	089-Welt erfolgreich vermessen.txt
058-Concorde-Flieger wollen Geld zurück.txt	090-Großbritannien untersucht Einschlagsgefahr.tx
059-Die Revolution stolpert voran.txt	091-Größter Tyrannosaurus rex wird ausgestellt.txt
060-Therapeutisches Klonen und Co.txt	092-18-Jähriger verbrannte sich die Augen.txt
061-Super-Sportler aus dem Labor.txt	093-Ein Pionier der Halbleitertechnik.txt
062-Deutsche Experten werten Mission als Erfolg.tx	094-Mir-Flug als Hauptgewinn.txt
063-Ältestes Tier mit Federn war doch kein Dinosaur	095-Das Problemkind.txt
064-Saurier zwischen Science und Fiction.txt	096-Therapeutisches Klonen im Tierversuch erfolgr

■ Viren - Krankheiten
 ▲ Technik - Kommunikation
 ◆ Freizeit - Vergnügen

◆ Literatur
 ◆ Forschung
 ● Transportmittel

■ Patente
 ○ Einzeltexte - Ausreisser
 ▼ Weltraum - Sonne

D.3 Document Map and Part of List for Task 4



033-Harry Potter im Legoland.txt	065-Neuer BSE-Skandal.txt
034-Gericht stoppt illegale Potter-Übersetzungen.txt	066-Schreckenskönig oder Schreckgespenst.txt
035-US-Jäger vor gefährlicher Hirnerkrankung bei V	067-Kosmisches Schlüsselloch.txt
036-VIAGRA.txt	068-Naturschauspiel verwirrt Himmelsgucker.txt
037-Gehirnentwicklung erfolgt noch bis zur Pubertät	069-Cluster-Sonden gehen in Position.txt
038-Begleiter wird aus Klinik entlassen.txt	070-Jubeln, Jammern, Schlafen.txt
039-Wer baut die UMTS-Handys von morgen.txt	071-Die Handhelds werden attraktive Ziele für Sch
040-Mehdorn ist der Totengräber dieser Technologi	072-Die Jugend war noch nie so frei und ungezwun
041-Stephen King auf neuen Wegen.txt	073-Carlsen macht Kompromisse.txt
042-Begeisterung, Staus und erste Augenbeschwer	074-Die Kontaktfreudige.txt
043-Schafft die Esa was der Nasa nicht gelingt.txt	075-Die Mir soll an die Börse.txt
044-Asteroid verfehlt Erde nur knapp.txt	076-Neurobiologe und Kunstsammler.txt
045-Die Übertragung ist nur über Mücken möglich.t	077-Die schöne neue Welt nimmt Formen an.txt
046-Sony steigt in den Handheld-Markt ein.txt	078-Sonnenstürme bedrohlicher als erwartet.txt
047-Mars - Erde 20.txt	079-Hubble entdeckt verführerischen Schleier.txt
048-Patient starb an Lassa-Fieber.txt	080-Neue Besatzung zur Mir unterwegs.txt
049-Der Vater des Mikrochips.txt	081-Wunderschön und geräumig.txt
050-Forscher entdecken Zentrum des Sextriebs.txt	082-Umstrittene Sonde Cassini passiert die Erde.txt
051-Atlantis-Crew verlässt Raumstation.txt	083-US-Experten warnen vor Epidemie.txt
052-Karasek knallhart.txt	084-Dinosaurier-Ei samt Embryo gestohlen.txt
053-Blutanalyse im Raumanzug.txt	085-Meister des Knochenpuzzles.txt
054-Chris Columbus verfilmt Harry Potter.txt	086-Die WHO gibt teilweise Entwarnung.txt
055-Knapp 600 Millionen telefonieren per Handy.txt	087-Flugmodul 5,6,7 und 8 suchen noch einen Na
056-Der erste Shopping-Assistent der ein Roboter is	088-Ebola-Virus.txt
057-Wegbereiter einer neuen chinesischen Literatu	089-Welt erfolgreich vermessen.txt
058-Concorde-Flieger wollen Geld zurück.txt	090-Großbritannien untersucht Einschlagsgefahr.tx
059-Die Revolution stolpert voran.txt	091-Größter Tyrannosaurus rex wird ausgestellt.txt
060-Therapeutisches Klonen und Co.txt	092-18-Jähriger verbrannte sich die Augen.txt
061-Super-Sportler aus dem Labor.txt	093-Ein Pionier der Halbleitertechnik.txt
062-Deutsche Experten werten Mission als Erfolg.tx	094-Mir-Flug als Hauptgewinn.txt
063-Ältestes Tier mit Federn war doch kein Dinosaur	095-Das Problemkind.txt
064-Saurier zwischen Science und Fiction.txt	096-Therapeutisches Klonen im Tierversuch erfolg

E Part of Speech Tags Used by the Semantic Refinement Module

Table A-1: Penn Treebank part of speech tags (excluding punctuation) [Bri95]

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential “there“
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PP	Personal pronoun
19.	PP\$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	“to”
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VCN	Verb, past participle
31.	VBP	Verb, non - 3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh - determiner
34.	WP	Wh - pronoun
35.	WP\$	Possessive wh-pronoun
36.	WRB	Wh - adverb

Attribute-Value Pairs

An attribute consists of the name of its base concept (which is equivalent to its name), a domain $G = [g_{\min}, g_{\max}]$ for the possible values, and a list of attribute values described by trapezoid fuzzy sets which are identified by a user-defined name. An attribute-value pair is a pair of the base concept (= attribute) name and a value name.

Example:

```
*FUZZY_SET
bandwidth
0 1000
low 100 200 300 400
high 500 800 1000 1000
```

```
<gmin> <gmax>
<z1> <z2> <z3> <z4>      ::= floating point number

<attribute value>         ::= <attribute value name> <z1> <z2> <z3> <z4>

<attribute value list>    ::= <attribute value> cr
                             {<attribute value> cr}

<attribute>               ::= *FUZZY_SET cr
                             <concept-id> cr
                             <gmin> <gmax>cr
                             <attribute value list> cr

<attribute list>          ::= <attribute> cr {<attribute> cr}

<attribute-value pair>    ::= <avp-id> <concept-id> <attribute value name>

<avp list>                ::= <attribute-value pair> cr
                             {<attribute-value pair> cr}
```

Syntactic Regular Expressions

A set of syntactic regular expressions can be defined by a list of regular expressions formed over the alphabet consisting of all possible part of speech tags (cf. appendix E) and the variable placeholders

/&1, /&2, /&3, /&4, /&5, /&6, /&7, /&8, /&9

For the syntax of the regular expressions itself refer to [HaBe92]. The slashes ‘/’ are token delimiters. For example, the regular expression

/&2/VB [DGNPZ] [(/RB)] */&1

matches patterns like /&2/VBD/RB/RB/&1 or /&2/VB/&1.

Example:

```
*SYNTACTIC_PATTERN
JJ-NN
/&1/&2
/&2/VB [ (/RB) ] */&1
```

```
<regex list>              ::= <regular expression> cr
                             {<regular expression> cr}
```



```

<syntactic pattern>      ::=  *SYNTACTIC_PATTERN cr
                             <pattern-id> cr
                             <regex list> cr

<syntactic pattern list> ::=  <syntactic pattern> cr
                             {<syntactic pattern> cr}

```

Rules: Weighted Element

This is the syntax for rule type \mathcal{R}_1 .

Example:

```
1_1    <message> 0.8
```

Note: 1_1 is a rule-id which is used internally in the semantic refinement module.

```

<weighted element rule>  ::=  <rule-id> <<element-id>> <weight>

<weighted element rule list> ::=  <weighted element rule> cr
                                   {<weighted element rule> cr}

```

Rules: Element Patterns

This is a special case of rule type \mathcal{R}_2 .

Example:

```
2_1    <Cisco> 0.8 <network> 0.2
```

```

<element pattern rule>   ::=  <rule-id>    <<element-id>> <weight>
                             <<element-id>> <weight>

<element pattern rule list> ::=  <element pattern rule> cr
                                   {<element pattern rule> cr}

```

Rules: Syntactic Concept Patterns

Syntactic concept patterns are sentence oriented. It is possible to choose between a simple heuristic matching where no syntactic pattern is given to specify a syntactically legal matching (indicated by a double asterisk *, rule type \mathcal{R}_3) and a matching which uses a grammar rule as defined by the indicated syntactic pattern (rule type \mathcal{R}_4). If the tags are omitted only the appearance of the concepts within a sentence are checked (without considering their grammatical category).

Example:

```

3_1    <send/VB, message/NN> 0.8 <receive/VB, message/NN> 0.5 * *
3_2    <send, message> 0.8 <receive, message> 0.5 * *
4_1    <send/VB, message/NN> 0.8 <receive/VB, message/NN> 0.5 VB-NN VB-NN

```

VB-NN must be a valid identifier of a previously defined set of regular expressions meant to recognise this tag combination.

```
<tag>      ::=  part of speech tag (cf. appendix E)
```

```

<tagged concept>          ::= <concept-id>/<tag>
<tagged concept list>     ::= <tagged concept> {,<tagged concept>}
<syntactic concept
pattern rule>             ::= <rule-id>    <<tagged concept list>> <weight>
                                <<tagged concept list>> <weight>
                                <pattern-id> <pattern-id>
                                |
                                <rule-id>    <<tagged concept list>> <weight>
                                <<tagged concept list>> <weight>
                                * *

<syntactic concept
pattern rule list>        ::= <syntactic concept pattern rule> cr
                                {<syntactic concept pattern rule> cr}

```

Rules: Document Oriented Element Patterns

In this rule type elements are matched against a complete document (rule type \mathcal{R}_2).

Example:

```
5_1    <send, message> 0.8 <receive, messages> 0.5
```

```

<element list>            ::= <element-id> {,< element -id>}
<element rule>            ::= <rule-id>    <<element list>> <weight>
                                <<element list>> <weight>

<element rule list>       ::= <element rule> cr
                                {<element rule> cr}

```

Rule Base

```

<rulebase>                ::= *CONCEPTS cr
                                <concept list>
                                <attribute list>
                                <syntactic pattern list>

                                *AV_PAIRS cr
                                <avp list>

                                *RULE_CW cr
                                <weighted element rule list>

                                *RULE_CWCW cr
                                <element pattern rule list>

                                *RULE_CN cr
                                <syntactic concept pattern rule list>

                                *RULE_DN cr
                                <element rule list>

                                *END

```

G Semantic Refinement Module: Rule Bases and Additional Experiments

The following sections contain the rule bases as used in the experiments according to the syntax defined in appendix F.

G.1 Rule Base “Aeronautics Patents”

```
*CONCEPTS
glider           := <glider,1.0,kite,1.0>
balloon          := <balloon,1.0>
airship          := <airship,1.0>
helicopter       := <helicopter,1.0>
rocket           := <rocket,1.0,missile,1.0>
rotor            := <rotor,1.0>
steering_device := <rudder,1.0,steering,0.8>
payload          := <payload,1.0, passenger, 1.0, freight,1.0, cargo,1.0,
                    luggage,0.8, weapons,1.0,missiles,1.0>
evacuation       := <evacuation,1.0,escape,1.0,rescue,1.0>
inflatable       := <inflatable,1.0>
slide           := <slide,1.0, ramp,1.0, slideway,1.0, chute,1.0,
                    system,0.8, device,0.8>
safety           := <safety,1.0>
escape           := <escape,1.0,retreat,0.8>
catapult         := <catapult,1.0>
launch           := <launch,1.0,take-off,1.0>
aircraft         := <aircraft,1.0>
light            := <light,1.0,lightweight,1.0,ultralight,1.0>
parachute        := <parachute,1.0,parachute-wing,1.0>
propulsion       := <propulsion,1.0>
motor            := <motor,1.0,engine,1.0,turbine,1.0>
tip              := <tip,1.0>
fin              := <fin,1.0,sail,0.8>
vertical         := <vertical,1.0>
wing             := <wing,1.0>
control          := <control,1.0>
*RULE_CW
1_01    <balloon>      0.3
1_02    <airship>      0.5
1_03    <helicopter>   0.4
1_04    <rotor>        0.1
1_05    <rocket>       0.5
1_07    <glider>       0.7
1_08    <parachute>    0.5
1_09    <launch>       0.1
1_10    <propulsion>   0.1
*RULE_CWCW
2_01    <balloon>      0.1    <airship>    0.1
2_02    <helicopter>   0.1    <rotor>    0.1
2_03    <glider>       0.2    <parachute> 0.2
*RULE_CN
```

3_01	<evacuation,slide>	0.1	<evacuation,slide>	0.1	*	*
3_02	<inflatable,slide>	0.1	<inflatable,slide>	0.1	*	*
3_03	<inflatable,slide>	0.1	<evacuation,slide>	0.1	*	*
3_04	<evacuation,slide>	0.1	<safety>	0.1	*	*
3_05	<evacuation,slide>	0.1	<escape>	0.1	*	*
3_06	<catapult,launch>	0.4	<catapult,launch>	0.4	*	*
3_07	<light,aircraft>	0.5	<light,aircraft>	0.5	*	*
3_08	<glider>	0.4	<light,aircraft>	0.5	*	*
3_09	<light,aircraft>	0.2	<balloon>	0.2	*	*
3_10	<light,aircraft>	0.2	<airship>	0.2	*	*
3_11	<motor,rotor>	0.2	<motor,rotor>	0.2	*	*
3_12	<vertical,fin>	0.3	<vertical,fin>	0.3	*	*
3_13	<tip,fin>	0.3	<tip,fin>	0.3	*	*
3_14	<vertical,fin>	0.3	<tip,fin>	0.3	*	*
3_15	<wing,tip>	0.2	<vertical,fin>	0.2	*	*
3_16	<aircraft,control>	0.2	<aircraft,control>	0.2	*	*

*END

G.2 Rule Base “Airplane Descriptions”

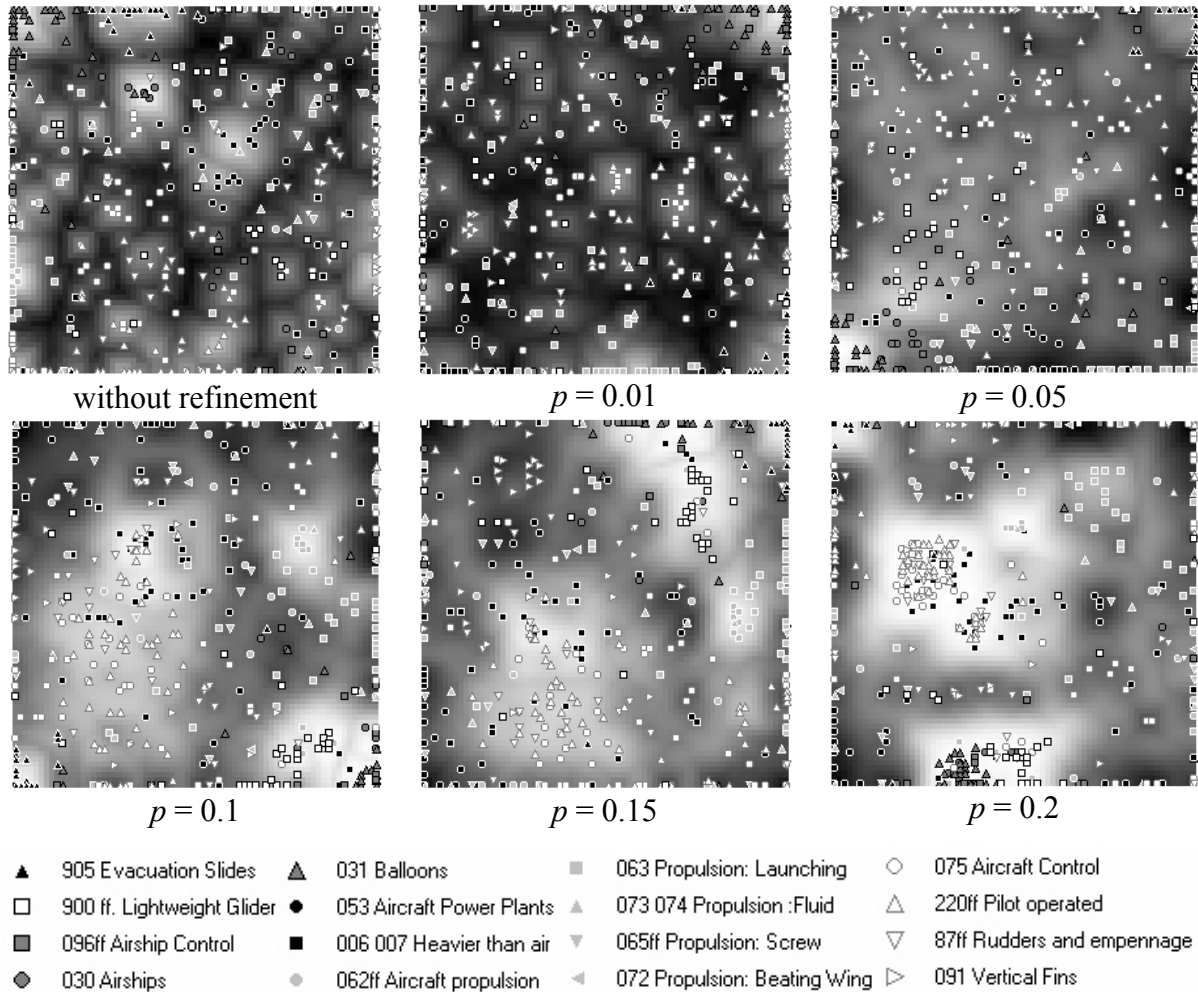
```

*CONCEPTS
wingspan      := <wingspan,1.0>
weight        := <weight,1.0,pounds,1.0>
*FUZZY_SET
wingspan
0 1000
small          0 5 25 30
medium         25 35 70 90
large          80 90 200 220
*FUZZY_SET
weight
0 1000000
small          0 500 10000 15000
medium         15000 30000 60000 90000
large          70000 80000 900000 1000000
*SYNTACTIC_PATTERN
AVP
/&1/&2
/&2/VB [DGNPZ] [ (/RB) ] */&1
/&2/IN/&1
*SYNTACTIC_PATTERN
NN-NN
/&1/&2
*AV_PAIRS
large_wingspan wingspan large
large_weight weight large
*RULE_CW
1_01      <large_wingspan>      1.0
1_02      <large_weight>        1.0
*END

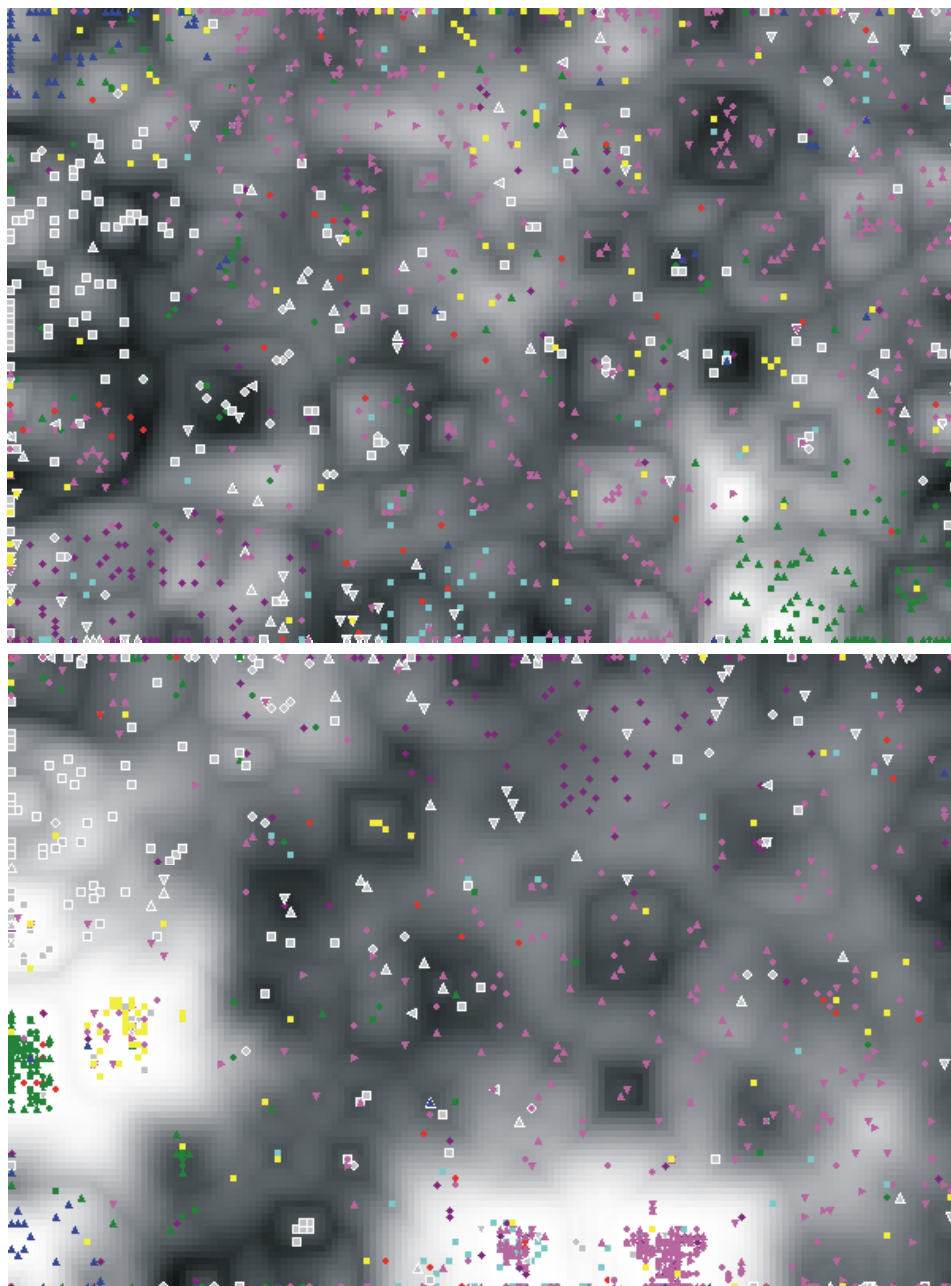
```

G.3 Series of Aeronautics Maps

The following series of document maps shows the effect of varying the movement length factor p of the semantic refinement component (cf. chapter 10.3.2). The document collection used is the corpus of aeronautics patent abstracts known from chapter 10.10.1. The corresponding rule base can also be found in appendix G.1. In all cases the maximal number of iteration, t_{\max} , is 10.



G.4 Map of Large Aeronautics Collection



- | | |
|---------------------------------|----------------------------------|
| ▲ 905 Evacuation Slides | ● 062 Aircraft Propulsion |
| ■ 900ff Lightweight Gliders | ■ 063 Propulsion - Launching |
| ■ 096ff Airship Control | ▲ 073 Propulsion - Fluid |
| ◆ 030 Airships | ▼ 065ff Propulsion - Screw |
| ▲ 031 Balloons | ◀ 072 Propulsion - Beating Wings |
| ◆ 053 Aircraft Powerplants | ◆ 075 Aircraft Control |
| ■ 006 Heavier than Air Aircraft | ▲ 220ff Pilot operated |
| | ▼ 087ff Rudders and Empennage |
| | ▶ 091 Vertical Fins |

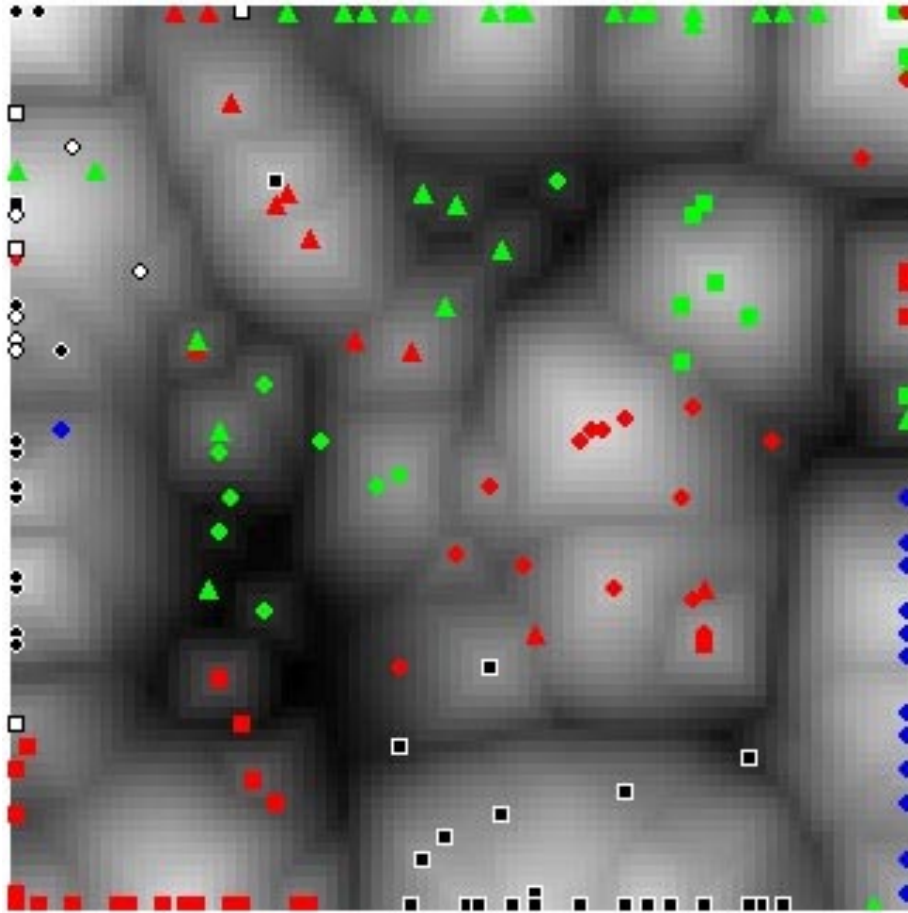
Pre-structured document map (top) and map refined with aeronautics rule base from appendix G.1 (bottom).

Parameters and Statistics of Map and Collection.

Note: The experiment has been performed on a Pentium II 350 MHz machine with 128 MB RAM and WinNT 4.0.

collection	name	aeronautics patent abstracts
	# documents	1,488
	avg. # words	123
indexing	size of vocabulary	4,170
	avg. # indexing terms / document	34
	weighting scheme	TF
	similarity measure	Euclidean
	dimension document space	40
semantic refinement	movement length factor (p)	0.1
	# iterations (t_{\max})	100
	# rules	28
	# resulting attractors	102,942
SOM	size of SOM	150×100
	training steps per document	20
times	basic map generation	47 min 15 sec
	semantic refinement	34 min

H Document Map of This Thesis



- Ch. 1: Introduction
- Ch. 2: Task-Model
- Ch. 3: Methods for Text Representation
- Ch. 4: Methods for Visualizing Similarity Data
- ▲ Ch. 5: Structuring Techniques
- Ch. 6: Basic Framework
- ▲ Ch. 7: DocMINER
- Ch. 8: Case Studies
- Ch. 9: Comparative Study
- Ch. 10: Extension of the Basic Framework
- Ch. 11: Conclusion