Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Fraunhofer
IAIS

Master Thesis

im Studiengang
Master of Science in Computer Science

Linear segmentation of ASR transcripts and text by topic

von
Peter Muryshkin

Erstbetreuer: Prof. Dr. Wolfgang Heiden
Zweitbetreuer: Prof. Dr. Peter Becker
Ext. Betreuer: M.Sc. Sebastian Tschöpel

Eingereicht am: 24.02.2011

This thesis is dedicated to my son David Rubin,

whose being is so much inspiring.

Abstract

The recent explosion of available audio-visual media is the new challenge for information retrieval research. Audio speech recognition systems translate spoken content to the text domain. There is a need for searching and indexing this data which possesses no logical structure. One possible way to structure it on a high level of abstraction is by finding topic boundaries.

Two unsupervised topic segmentation methods were evaluated with real-world data in the course of this work. The first one, TSF, models topic shifts as fluctuations in the similarity function of the transcript. The second one, LCSeg, approaches topic changes as places with the least overlapping lexical chains.

Only LCSeg performed close to a similar real-world corpus. Other reported results could not be outperformed.

Topic analysis based on the repeated word usage models renders topic changes more ambiguous than expected. This issue has more impact on the segmentation quality than the state-of-the-art ASR word error rate.

It could be concluded that it is advisable to develop topic segmentation algorithms with real-world data to avoid potential biases to artificial data. Unlike evaluated approaches based on word usage analysis, methods operating with local contexts can be expected to perform better through emulation of semantic dependencies.

Acknowledgements

I wish to thank all people that have made this thesis possible. My colleagues at the Fraunhofer facility gave me a warm welcome and good mentoring which led me to the research area presented in this thesis. Fraunhofer is a great place to write a thesis if you are looking for an environment providing a perfect combination of science and bleeding-edge technology.

It is very important to mention all of the people whom I do not know personally but we all benefit daily from their hard work. These are the innumerable open source community enthusiasts who gave us Eclipse, LaTeX and so much more. I thank also Stefan Macke for providing the LaTeX template which I could extend and adjust for use with this thesis, as well as the team behind Aigaion, an excellent online bibliography management tool.

I thank my parents for teaching me English in my childhood, probably contributing an invaluable part to the fact I was able to write this thesis in English. I also greatly appreciate Terry's effort which she invested for proofreading this thesis, and communication studies literature advice given by Frank.

Furthermore, I would also like to express my appreciation to the participants of the online survey dealing with annotating topic boundaries for the DiSCo corpus, and to Verena for providing the web template for the survey tool.

This work also would not have been possible without the moral support and kind incitements provided by Eve and Kirsten in the last months. Eve has also annotated large parts of the DiSCo corpus with reference boundaries.

Eidesstattliche Erklärung

Name: Peter Muryshkin

Matrikelnr.: 9003600

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Peter Muryshkin

Sankt Augustin, den 24.02.2011

Contents

List of Abbreviations **VII**

List of Figures **VIII**

List of Tables **X**

1. Introduction **1**
1.1. Research context 1
1.2. Aim and purpose 2
1.3. Thesis overview 2

2. Background **4**
2.1. Topic definitions 4
2.2. Applications of topic segmentation 6
2.3. Problem description 8
2.4. Topical structure indicators 9
2.4.1. Transcript-intrinsic clues 10
2.4.2. Transcript-extrinsic clues 12
2.4.3. Conclusion 13
2.5. System landscape 13

3. State of the art **15**
3.1. Measuring the quality of topic segmentation 15
3.1.1. Precision, recall and the F-score 16
3.1.2. Beeferman error metric (P_k) 17
3.1.3. WindowDiff error metric (WD) 18
3.1.4. Pr_{error} metric 19
3.1.5. Topic closeness measure (TCM) 19
3.2. Conventional corpora overview 20
3.3. Classification of known approaches 21
3.3.1. Common preprocessing steps and general algorithm layout . . 21
3.3.2. Methods treating lexical cohesion/lexical similarity 24
3.3.3. Clustering methods 29
3.3.4. Dynamic programming 32

3.3.5. Machine learning 36
3.4. Discussion 43
3.5. Related work 47

4. Applied methods **48**
4.1. Used corpora 48
4.2. Method selection 49
4.3. TSF 51
4.3.1. Baseline implementation 51
4.3.2. Baseline results 52
4.3.3. Conclusion 53
4.4. LCSeg 54
4.4.1. Baseline implementation 54
4.4.2. Baseline results 55
4.4.3. Analysis 56
4.4.4. Experiments 62
4.4.5. Conclusion 63

5. Implementation details **65**
5.1. Algorithm analysis 65
5.2. Architectural concepts 66
5.2.1. Middleware 66
5.2.2. Models 67
5.2.3. Controllers 68
5.2.4. Views 69
5.3. Performance 70

6. Conclusion and future work **71**
6.1. Summary 71
6.2. Conclusion and future work 72
6.3. Further reading 73

Bibliography **74**

A. MPEG7/XML example (a fragment) **ii**

B. Best-performing topic segmentation methods **iii**

C. Online topic annotation tool **iv**

D. A configuration example for topic segmentation execution **v**

Index **vii**

List of Abbreviations

ASR	Automated Speech Recognition
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DiSCo	Difficult Speech Corpus
FN	False Negative(s)
FP	False Positive(s)
GLSA	Generalized Latent Semantic Analysis
HMM	Hidden Markov Model
HTMM	Hidden Topic Markov Model
LCA	Local Context Analysis
LCF	Lexical Cohesion Function
LCP	Lexical Cohesion Profile
LDA	Latent Dirichlet Allocation
LM	Language Model
LSA	Latent Semantic Analysis
LUT	Language Understanding Toolbox
NLP	Natural Language Processing
PLSA	Probabilistic Latent Semantic Analysis
PMI	Pointwise Mutual Information
SVD	Single Value Decomposition
TDT	Topic Detection and Tracking
TP	True Positive(s)
VMP	Vocabulary Management Profile
WER	Word Error Rate

List of Figures

2.1. Indicators of topical structure 10
2.2. *AudioMining* system component diagram 14

3.1. Workwise of the P_k algorithm 18
3.2. Demonstration of the weakness of P_k 18
3.3. Research on topic segmentation 22
3.4. General layout of an unsupervised topic segmentation algorithm . . 23
3.5. Example of a simple semantic network 24
3.6. Example of an LCP graph 25
3.7. TextTiling example 26
3.8. Lexical chains 28
3.9. Topic dotplots 31
3.10. Anisotropic diffusion applied to a sentence distance matrix 33
3.11. Spanning graph of a text 34
3.12. General layout of a supervised topic segmentation algorithm 37
3.13. Log ratio of a trigram and a long-range model 38
3.14. Comparison of the HMM and the AHMM approaches 40
3.15. Linear vs. non-linear classification 41
3.16. Example of a decision tree 43
3.17. Research on topic segmentation reporting the best results 44

4.1. Human perception of topic segmentations 49
4.2. Details of TSF algorithm 51
4.3. An example of TSF execution 53
4.4. An example of TSF execution with final segmentation 53
4.5. An example of LCSeg execution 56
4.6. An example of LCSeg execution (chains) 57
4.7. LCSeg: boundary-cutting chains 59
4.8. LCSeg: too long chains 61

5.1. Interface and class diagrams for transcript providers 67
5.2. Class diagrams for topic segmentation asset and segment boundary . 68
5.3. Sequence diagram of the topic segmentation pipeline 69
5.4. Class diagrams for the charting functionality 69

C.1. Online topic annotation tool iv

List of Tables

3.1. Corpora overview 21
3.2. Topic segmentation methods with $P_k \leq 10\%$ 45
3.3. Topic segmentation methods tested with ASR transcripts, $P_k < 25\%$ 45

4.1. Baseline TSF results 52
4.2. Baseline LCSeg results 55
4.3. LCSeg: example of false negatives 58
4.4. LCSeg: misses and matches(1) 58
4.5. LCSeg: misses and matches(2) 58
4.6. LCSeg: analysis of boundary-cutting chains 60
4.7. LCSeg: Experiments proposals 62
4.8. LCSeg: Experiments results 63
4.9. LCSeg: Experiments details 64

B.1. Details on the best-performing topic segmentation methods. iii

1. Introduction

This introductory chapter explains the environment of this thesis and presents its background and aim. At the end a short chapter overview is given.

1.1. Research context

This work has been conducted at the *NetMedia* division of the *Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)*[1] to support the work of the audio team[2].

Due to their articles of association, all Fraunhofer facilities focus research on applied sciences. In the case of the IAIS this is applied computer science. This enables the NetMedia division to develop and provide innovative methods and systems for digital media presentation and data management along with participating in real-life projects through collaboration with industrial partners.

The inevitability of the need for innovative approaches to deal with digital media appears quite natural regarding the dramatic growth rate of worldwide multimedial data. As estimated by [LV00], total worldwide TV and radio broadcasting produces about 100 petabytes of original content annually. This corresponds to roughly 65 million hours of TV and 48 million hours of radio broadcast. Additionally, [Gan07] reports 95% of all existing digital data to be unstructured (i. e. not indexed for a quick search) and predicts its total size to grow from 161 to 988 exabytes between 2006 and 2010.

As is generally known, available storage capacities get cheaper steadily year by year and digital production has become dominating. This means that all the data can be easily archived, for example, for reuse purposes (only 25% of total TV broadcast is new content [LV00]). The task of managing and searching such vast archives can apparently become intractable. Searching multimedial data is more complex than searching text and effective processing of search queries of this type is a big challenge.

[1] Website: http://www.iais.fraunhofer.de/netmedia.html
[2] Website: http://mmprec.iais.fraunhofer.de/speech-audio

To provide better exploration of multimedial data, the NetMedia audio team introduced the audio speech recognition (ASR) system *AudioMining*. It bases on algorithms for speech recognition and produces basically a time-aligned textual transcript in MPEG7/XML format with other metadata of what was said in an archived TV or radio broadcast or also any other dataset containing speech as audio record. Such transcripts allow, for example, instant searching for a term instead of time-consuming manual transcribing, indexing or searching by skimming the audio track of interest. Specifically, a search input returns transcript matches with corresponding time codes which can be then directly accessed in a player software. An exemplary fragment of an MPEG7 news transcript is given in Appendix A.

1.2. Aim and purpose

This thesis deals with methods regarding topic-based segmentation of news broadcast transcripts created by *AudioMining*. Currently these transcripts contain no information about topic boundaries in the audio stream. However, this information can be very useful (i. e. for better media navigation). Segmentation of a transcript by topic is also an important processing pre-stage of topic recognition which allows for content recommendation or an ASR reiteration with a more specific vocabulary to improve the quality of speech recognition.

The goals of this thesis are to:

- identify state-of-the-art methods for topic segmentation of text streams, ideally suitable also for ASR transcripts;
- choose, implement and compare one or two of identified unsupervised methods;
- provide a new software component for *AudioMining* to tackle the task of topic segmentation.

1.3. Thesis overview

This chapter is followed by Chapter 2 (Background), which delves into more detail about the problem definition and indicators of topic segmentation. Finally, conceivable applications and system environment of this work are described.

Chapter 3 (State of the art) discusses quality measures for topic segmentation and presents conventionally used test data sets, followed by a broad overview of previously elaborated segmentation methods.

Chapter 4 (Applied methods) describes data collections used in the course of this work and after a short discussion studies two previously selected topic segmentation methods.

In Chapter 5 (Implementation details) the software design used to implement both methods in Chapter 4 is presented.

Finally, Chapter 6 (Conclusion and future work) summarizes the results and insights gained during this work, concluding with a prospect on the future work and information for further reading.

2. Background

This chapter provides background information related to topic segmentation. Definitions of the term topic in different disciplines are explored, followed by considerations on what the topic segmentation problem is and which challenges it rises. Finally, topic structure indicators, i.e. discourse and media features which could give a hint about topic boundaries are presented. The chapter concludes with some general details on the technical background of this work and the surrounding system landscape.

2.1. Definitions of the term "topic" and how it is used in this work

The notion of the term *topic* is quite intuitive. It implies some subject which can be discussed. This section contains formal definitions and sheds more light on this term.

Etimology and common definitions

The English word *topic* originates from its Latin predecessor *topica*, which in turn comes from the Greek $\tau o \pi \iota \kappa \acute{o} \varsigma$ which originally means *related to a (common) place* [Web08][1]. In his *Topica* Aristotle mentions topics to be parts of, or places in an argument [Ari89].

In modern English, a topic is referred to as

- "a subject discussed in a speech, essay, thesis, or *part of discourse*";
- "a *subdivison* of a theme, thesis or outline";
- "the subject of a discourse or of a section of a discourse" [Web08].

[1] The Greek word $\tau o \pi o \varsigma$ stands for *place*.

Linguistics

In linguistics a topic is "a word or phrase in a sentence, usually providing information from previous discourse or *shared knowledge*, that the rest of the sentence elaborates or comments on" [Web08].

Specifically, the term topic belongs to the linguistic concept of *topic-comment*, meaning "the phrase in a discourse that the rest of the discourse is understood to be about." In English, the topic is normally indicated by the subject in a sentence (what is being talked about). The comment (also *focus* or *rheme*) corresponds to what is being said about the topic [Giv83].

Other languages can have other ways to emphasize the topic-comment structure. For example, in Japanese a special postposition *-wa* [NN94] is used, and the Korean language has a dedicated part of speech called *topic particle*. Due to this fact, a classification of languages to topic-prominent and subject-prominent ones was introduced [Li76].

A closely related linguistic term to mention is *lexical cohesion*, explored by [HH76], meaning grammatical and lexical relationships in a text which hold it together, giving it a meaning. As we will see, lexical cohesion is a feature widely exploited by many segmentation algorithms.

Communication studies

While any meaningful text can be expected to be related to some subject (e.g. consisting of at least one topic), every single-topic segment can also have its own structural pattern, including recursion in the form of introduced subtopics, which are also topics in their turn [PS83]. Discourse analysis also refers to linguistics by using the terms *cohesion* and *cohesive links* (anaphoras) to denote the "ties and connections which exist within texts" beyond the sentence level [Yul96]. Furthermore, the structure of discourse can be also predicted to a high extent if a conversation is conventional and institutionally and/or implicitely predefined (e.g. in call center dialogues [KPRS09], job interviews or even love professions [Aue96]).

Research terminology in computer science

Scientific publications related to topic segmentation do not always use the same terminology. Terms like *story segmentation* or *determing story/topic boundaries* or *discourse segmentation* are used as well. However, [Rey98] proposes to reserve the term *discourse segmentation* for a hierarchical analysis.

So, [ALJ00] defines a story to be a "topically cohesive segment of news that includes two or more declarative independent clauses about a single event." A topic is then "an event or activity, along with all directly related events and activities," which is tightly related to the focus on news broadcasts taken by the DARPA TDT research program (see [ALJ00] for more details) [2]. The term *event* is meant to be an event featured in the news, unlike the notion of *speech event* in communication studies [Yul96]. Another relevant term is *topic shifts* [Rey98; Pur11] which means changes of topicality in a document.

An important distinction to consider is between the terms *topic segmentation*, *topic labeling* and *topic detection*. While topic segmentation means finding boundaries between topically coherent regions in the text, topic labeling means associating these regions with specific categories (e.g. sports or cooking). Finally, topic detection and tracking stands for finding occurrencies of one specific topic.

On the contrary, the two following terms will be used as synonyms in the course of this work: *transcript* and *document*, for transcripts can be handled in terms of information retrieval as documents.

Finally, the term *asset* will be used to define a single task of topic segmentation in general, related depending on the context to the transcript or to its multimedial source.

Conclusion

As we can see from the topic definiton and its role, topics appear to be important discourse elements because they provide semantic boundaries, or segments of discourse. Understanding language-specific differences and ways to communicate could be probably useful for developing well-performing topic segmentation algorithms.

2.2. Applications of topic segmentation

Segmenting large media collections by topic can be applied to a broad variety of tasks. This section gives an overview of possible applications.

[2] TDT stands for *Topic Detection and Tracking*, this research effort went on in 1998 – 2003. See also http://ciir.cs.umass.edu/projects/tdt .

Improving the end-user experience

As pointed out in the first chapter, vast digital media archives containing millions of hours of visual and speech information have become available, annualy growing by more than 110 million hours as of 2000 [LV00]. Topic-based search would be an undoubtedly time-saving feature.

A topic-labeled archive also enables indexation and searching on a more abstract level, which allows for more precise and fast search responses of a search engine.

Another application for the end-user could be recommendation of topic-related content or tracking development of some topic over a large time scale (e.g by a media observer).

Domain-specific information retrieval

Considering domain-oriented patterns in communication [Aue96] allows investigation of more application-specific tasks than is possible through general topic segmentation.

Examples of such applications are meeting and decision analysis [DR07] or data mining in lectures [CT10; SM08] to implement innovative ways to represent this data.

Advanced usage

Due to the fact that topic segmentation applies a logical structure to an ASR transcript, the topic segment boundaries can also be used as a prerequisite for further processing. Having large transcripts split into smaller and topic-coherent fragments enables designing algorithms in the *divide and conquer* approach [Cor02].

A special case is the task of text summarization [Hea97; Cho02]. It is usually based on creating representative shortened versions of text parts like chapters or paragraphs. Since ASR transcripts do not possess any logical structure, summarization by topic appears to be a feasible way to tackle this task.

Another example brought by [LTM10] along with text summarization is *anaphora resolution* [Mit02] and [Koz93], for anaphora have a limited appearance scope (e.g. up to 17 sentences [Mit99]). Analysis of smaller segments might therefore improve the quality of anaphora resolution.

Finally, *AudioMining* and assumably other ASR systems can operate with topic-based vocabularies along with the common data set. However, specific vocabularies

cannot be applied to multitopic assets. So after segmenting a transcript by topic a better utterance mapping can be done, improving the speech recognition quality by doing a reiteration. Providing a very specific vocabulary enables a recursive segmentation and recognition steps then. Vice versa, a specific vocabulary could be derived from a single-topic segment [Rey98, p.143].

2.3. Problem description

Task formulation

There seems to be no common formal definition of what topic segmentation stands for. However, the term itself is implicitely clear and apparently intuitive. [Pur11] describes the task of topic segmentation as following: "dividing single long recordings or transcripts into shorter, topically coherent segments".

Due to the recursive structure of discourse [PS83] a distinction between linear and hierarchical segmentation can be drawn. The discussion in [Pur11] points out that though formal models of dialogue exist, no hierarchical topic segmentation might be possible which would not depend on the perception context. However, [Cho02; SM08] and [Eis09] proposed algorithms for hierarchical segmentation by topic.

There is also a difference due to inner coherence of transcripts. There are more or less topically coherent transcripts composed of subtle subtopics (e.g. a discussion) and there are data streams consisting of topically unrelated pieces, for example, news broadcasts.

This thesis deals with parsing of less coherent ASR transcript files produced by *AudioMining* and enriching them with topic boundaries information using linear segmentation algorithms.

Complexity of spoken language

The segmentation itself is not an exactly defined task. So, human annotators tend to propose different topic boundaries [ALJ00; Bal04], with deviation growing towards transcripts containing deeper domain knowledge [GNP05]. An example of different boundaries proposals made by different humans will be presented later on in 4.1.

A further problem beyond lack of structure in ASR transcripts is that as demonstrated by [MW98], the notion of sentences being natural for written language is inapplicable in the domain of spoken language (e.g. due to partly incoherent syntax structutures in both domains).

From this point, the topic segmentation problem or input data will be considered as "hard" if the input data has no clear topic changes (e.g. a lecture or a talk show), and not hard if there are sharp topic shifts (e.g. a news broadcast or an artificially compiled document).

Moreover, topic segmentation can be complicated by language diversity. It is an important feature of human communication, which makes our interaction less error-prone. This is possible through using figures of speech (*tropes*) or paraphrasing [LLS95]. Beyond that, different vocabularies depending on, for example, the situation or a person's education, can be introduced.

Input quality

Finally, no ASR system is perfect and its output always has a word error rate. This means that quality of proper matching between utterances and corresponding words depends on the *language model* (word probability distribution)[PC95] as well as original audio record quality. In terms of analyzing asset topicality an erroneous ASR input can influence the segmenter in a bad way by introducing off-topic terms which are extraneous to the context. However, in the case of broadcast domain, especially news, there is normally a flawless audio signal quality, and possible ASR problems might relate rather to untrained or dialect speech.

2.4. Topical structure indicators

Any algorithm we would design to look for topic boundaries would need some formalizable clues. Conversely, there might be negative clues (e.g. evidences that a text segment contains *no* topic boundary).

While conventional texts are usually structured in chapters and paragraphs, an ASR transcript or at least its textual part possesses no textual structure beyond the word level. However, if the task of topic segmentation is often handled as a binary classification problem of boundaries between text elements as being simulatenously also a topic boundary or not, we can do the same using transcript metadata. This means, we can examine boundaries between transcript segments defined through long silent pauses or intermediate non-speech segments. However, we would miss a sudden topic shift in a block of more or less continuous speech if we do not consider other features.

This section gives an overview of topic shift clues regarding different levels of perception and complexity (see also Fig.2.1).

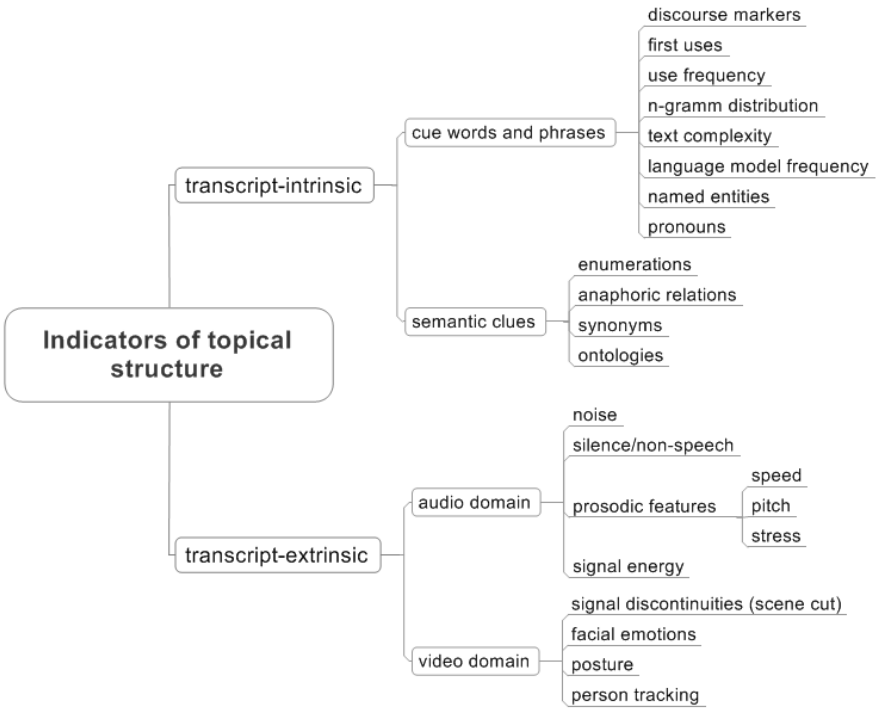

Figure 2.1.: *Indicators of topical structure and possible topic shifts.*

2.4.1. Transcript-intrinsic clues

Cue words and phrases

Usually single words of a transcript are normalized by a language-specific *stemming algorithm* which reduces the noise of token variety caused by morphological inflexions as preprocessing step (e.g. "going" is reduced to "go"). However, ASR transcripts usually contain no clues to recognize words to be part of a speech which is needed by advanced stemmers based on sentence analysis.

As pointed out by [GS86], some words or phrases do not transport information related to discourse subject but indicate changes in the discourse. Though language-specific, such *discourse markers* can be domain independent, like *actually* or *now*. [Rey98] shows benefits of elaborating domain-specific markers, e.g. *good morning* in the broadcast news domain. A drawback of this indicator is the necessity to derive discourse marker lists manually. In addition, there is enough room for misinterpretations, especially in transcripts where no punctuation evaluation is possible (*it is now affordable...*).

[You91] assumed that introduction of many new words could occur along with beginning of a new topic ("first uses").

Complementary to the first uses indicator, frequently used but not uniformly distributed words might belong to the same topic segment. However, this assumption does not take into account the existence of homographs, e.g. words which have the same spelling but different meanings (*lead pipes, lead the team*).

On a more abstract level, looking for word repetitions is a special case of text analysis. This means that a single word can be seen as a unigram , two as a bigram and so on. It is obvious that occurrence of an n-gram decreases with its growing length (therefore we use word combinations to get better search results from a search engine). So a cluster of repeated words or phrases could correspond to a topically coherent text segment. A more specific analysis could include extracting local terminologies [JK95] or named entities to focus on more important words. A negative clue in this case is the observation that new topics rarely begin with pronoun usage. Similarly, conjuctions or conjunctive adverbs (*and, however*) would neither occur as topic boundary.

Further improvement of previous concepts regarding repeated words is to consider general word frequencies in the language context. This allows for weighting word occurrences by corresponding probability to emphasize rareness of topic-specific words. However, a specific language or even domain model is needed then.

Filtering or simply ignoring frequent words containing less subject information is generally regarded as *stop word removal* .

One another indicator of changed topic could be text complexity measured by word and sentence length. However, with ASR transcripts this is not possible without previous sentence recognition due to the absense of punctuation.

Semantics

Recognition of semantic level clues involves processing content “meaning,” which requires deeper language understanding than operating just on tokens and their patterns.

A very simple example of a negative clues in this case are enumerations, like *firstly, secondly, thirdly.* It is unlikely that a new topic would start inside an enumeration [LP07].

More demanding to recognize is probably absence of anaphoric relations between two subsequent transcript segments: this could be an indicator for a topic shift.

Further, while use of synonymy might be misleading since synonyms do not necessarily appear in the same context through a multi-topic transcript, usage of language

ontologies might be helpful, as an ongoing discussion about the same topic might show close matches of semantically related words.

2.4.2. Transcript-extrinsic clues

Since speech is a real-world phenomenon which occurs in space and time, textual transcripts are not the only possible information source. Recent advances in audio and image processing, as well as growing computing power of affordable hardware, allows practical applications which consider multiple sources to get more clues about topic shifts. *AudioMining* produces transcripts which contain only some limited meta data; however, also the original media is archived after processing and can be therefore easily accessed.

Audio domain

The most simple potential (but not definite) topic boundaries are speech discontinuities like silence, non-speech segments or speaker changes; all these features are recognized by *AudioMining* and recorded along with word utterances. All of these features can be also incorporated by the same topic. Time alignment of all utterances allows also evaluation of the speech speed, for changes in speech speed might be a clue for topic changes [KIO96; SSHTT00; MPBG07].

Audio signal analysis involves more complex processing and might provide information like background noise (e.g. change of noise through a scene cut or accompanying music) or prosodic features (i.e. *how* a speaker produces his utterances). This includes signal energy or sudden changes in pitch and stress. There are also domain-specific clues in broadcast records, like jingles demarcating programs or their parts.

Visual domain

Transcripts which are made from audiovisual media can be analyzed using additional information from the corresponding video tracks. Topic changes can be recognized then through a range of features from simple signal discontinuities like scene cuts to complex clues involving advanced image processing like a person's facial emotions and posture or person tracking [May98; DR07; Pou09; GPHJ09].

Temporal domain

In the special case of TV broadcast news additional domain-specific knowledge can be used to hypothethize topic boundaries. So, while approaching the end of a segment with its length corresponding to an average story, the probability of the topic boundary around it would dramatically increase [May98].

2.4.3. Conclusion

We can conclude that there is a broad variety of features indicating possible changes of topicality, however none of them gives enough evidence. Analysis of these features might require sophisticated methods based on deep knowledge of linguistic phenomena, or in the case of some transcript-extrinistic clues, also complex signal processing. For more detailed information and further references please see [Rey98] and [Cho02].

2.5. System landscape

This section gives a short overview on the *AudioMining* system landscape and its components. While one of the goals of this thesis is to deliver a new software module, any software component has usually an execution context. Figure 2.2 is a top-level component diagram of the *AudioMining* system.

AudioMining impements the *service layer* [Fow02] architecture design pattern providing modular functionality, which wraps domain-specific data models and algorithms. The service layer incorporates a set of webservices providing execution of specific processing steps. Logically seen, the new module for topic segmentation fits to the section of structural analysis webservices. It would consume the output of for example the *segmentation* webservice and produce a MPEG7 file enriched with information about topic boundaries.

The system takes a multimedia asset as input supporting a broad variety of common audio and video formats and processes this data to create metadata XML files containing speech transcript along with a set other features. However, the most essential information for topic segmentation along with the lexical layer are silent or non-speech segments as well as the time alignment of the transcript. Provided timecodes for each utterance, a topic boundary set has a very simple notation being just a timestamp sequence containing segment starting points.

Due to the service-oriented architecture design some new services related to natural language processing are encapsulated in an *AudioMining* -external software package

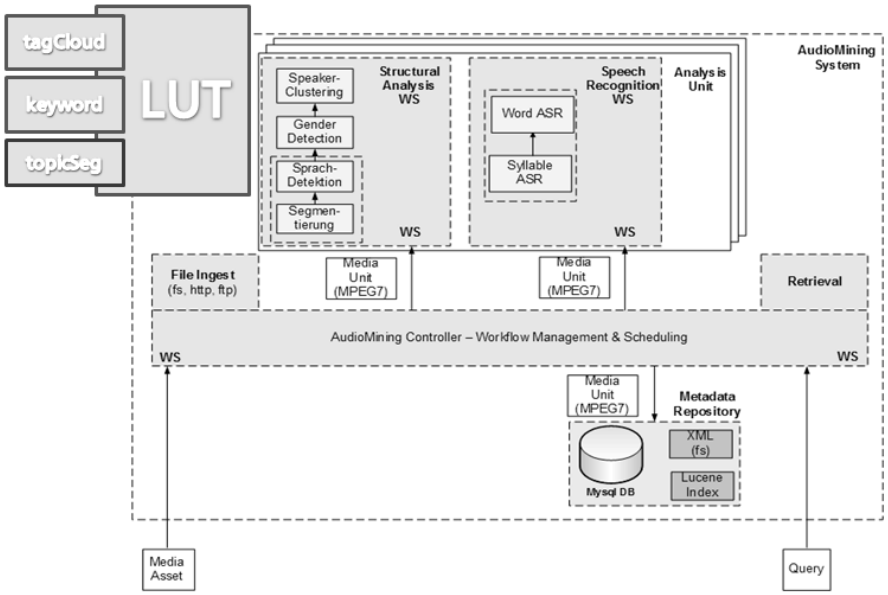

Figure 2.2.: *Component diagram of the AudioMining system. The service-oriented architecture provides on different abstraction levels. A new module providing the service for topic segmentation will be integrated into the LUT services set. ©2010 Fraunhofer IAIS*

LUT (language understanding toolbox). Topic segmentation will be realized as a new webservice placed there and providing additional rich annotation of ASR transcripts with topic boundaries.

3. State of the art

This chapter discusses applicable methods for measuring the quality of a topic segmentation algorithms and gives an overview on conventional test data sets, or corpora, in this research area. Finally, a review of publications on topic segmentations in the two past decades is given.

3.1. Measuring the quality of topic segmentation

As already mentioned in 2.3, segmentation of a discourse by topic can be very ambiguous. [Bal04] reports human annotators to produce *self-normalized* segmentations (i.e. there is some kind of psychological expectation for topic changes). This means that if no clear topic shift comes, a human annotator would feel a need to find one which results in relatively uniformly distributed topic boundaries across the transcript (see also 4.1).

For this reason both [ALJ00] and [Bal04] propose finding a *gold standard* segmentation by doing segmentation through a number of annotators and finding subsequently a good average which all participants can agree upon.

In further discussion, real, actual, or true boundaries denotate such topic boundaries which were or could be found through establishing such gold standard or generally human "common sense" for the first discourse hierarchic level where more than one boundary can be determined. First discourse hierarchic level means that no boundaries would be recognized if the whole asset deals with one general topic and respectively, taking boundaries of deeper subtopics means considering hierarchical segmentation which is beyond the goal of this work. To differentiate from "real" topic boundaries, boundaries found by an algorithm will be mentioned as proposed, supposed, assumed or hypothethized boundaries. Potential topic boundaries are then boundaries given by the existing structure of an asset (e.g. ends of sentences or, in the case of ASR transcripts, more likely blocks of continuous speech).

3.1.1. Precision, recall and the F-score

In information retrieval it is common to measure the quality of an algorithm by determining its *precision*, *recall* and their harmonic mean, called also *F-score* or *F-measure* [MKSW99].

Consider potential topic boundaries to be an instance set M for the classification problem task. Further on, there are two classes, potential topic boundaries which are also actual topic boundaries (C_t) or supposed but not actual (C_f). A classifier should partition the set of instances in two distinct sets M_1 and M_2: $M_1 = \{x|x \in C_t\}, M_2 = \{x|x \in C_f\}$ whereas $M_1 \cap M_2 = \emptyset$.

True positives (TP) are then real topic boundaries marked as topic boundaries, false positives (FP) are non-topic boundaries marked as topic boundaries, and false negatives (FN) are missed actual topic boundaries which the algorithm fails to recognize.

Precision P is then the ratio of the number of true positives and the total number of instances which the algorithm believes to be topic boundaries ($|M_1|$) (i.e. sum of true positives and false positives). Low precision corresponds to oversegmentation. High precision means low amount of false boundaries.

$$P = \frac{TP}{TP + FP} \tag{3.1}$$

Respectively, recall is defined as the ratio of the number of true positives divided by the total number of real topic boundaries ($|C_t|$, i.e. sum of true positives and false negatives). Low recall of an algorithm means that it fails to locate real topic boundaries and misses many of them. High recall means good matching of them.

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

Finally, the F-score is calculated as the harmonic mean of precision and recall.

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{3.3}$$

An F-score of 1 or 100% and 0 or 0% correspond to the best and respectively the worst possible rating.

However, as pointed out by [BBL99], this metric does not endorse finding a topic boundary which is *close* to a real boundary, considering inputs in a binary way to be just correct or wrong. Still, an algorithm which is able to find close matches should

be considered as performing better than some other one proposing boundaries (e.g. in a random way). The F-score measure would score them equally and therefore does not provide meaningful comparison of topic segmentation algorithms.

For this reason and due to the fact that precision and recall are very well-known in the information retrieval domain, we can introduce adapted metrics P^*, R^* and the $F^* - score$. We deal then with false positives (FP), false negatives (FN) and true positives (TP) in the same way as in normal case but allowing good scoring for matches coming close to the reference through a tolerance window.

3.1.2. Beeferman error metric (P_k)

[BBL99] proposes a probability-based metric P_k which is the most broadly-used metric to evaluate topic segmentation algorithms. For every two points of an asset an average probability can be determined if a segmenter separates them by a boundary or not. A good scoring is close to 0 (corresponding to zero error probability).

P_k is computed with the help of the indicator function δ (Eq. 3.4). With an arbitrary segmentation S and a sliding window of fixed width k and for each position of the window start i and end j, δ is then defined as the following (the following formulae and more explanation on the metric derivation are to find in [Pur11]):

$$\delta_S(i,j) = \begin{cases} 1 & \text{if segmentation } S \text{ assigns } i \text{ and } j \text{ to the same segment;} \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

The indicator function works as the following. Consider a hypothetical segmentation H produced by the algorithm and a "real" reference segmentation R given by a human. For each position of the sliding window we can tell if there is a match or mismatch between H and R (match is defined for the case if in both H and R is at least one or zero boundaries to see through the sliding window). Formally, the XOR operator $\oplus$ on δ_H and δ_R returns 1 if and only if there is a mismatch between H and R. The P_k is then the number of such mismatches found through summarizing δ for all window positions divided by the number of windows.

$$P_k = \frac{\sum_{i=1}^{N-k} \delta_H(i, i+k) \oplus \delta_R(i, i+k)}{N-k} \tag{3.5}$$

Fig. 3.1 illustrates the way P_k is calculated.

[Pur11] points out that P_k can be also calculated as following

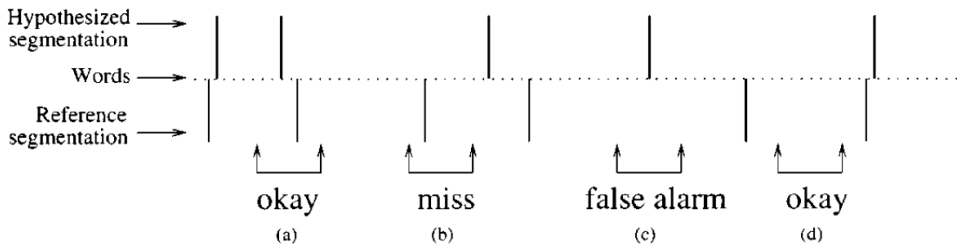

Figure 3.1.: *Workwise of the P_k algorithm. The sliding window positions (a) and (d) are not scored because their evaluated segmentation matches the reference, and the sliding window positions (b) and (c) are (negatively) scored. [BBL99]*

$$P_k = P_{miss} + P_{falsealarm} \tag{3.6}$$

This might be useful for evaluation of research results which refer to these metrics instead of P_k.

3.1.3. WindowDiff error metric (WD)

P_k is undoubtedly more suitable for measuring the quality of a topic segmentation algorithm than the F-score. However, [PH02] gives an example where P_k fails to penalize false alarms (Fig. 3.2).

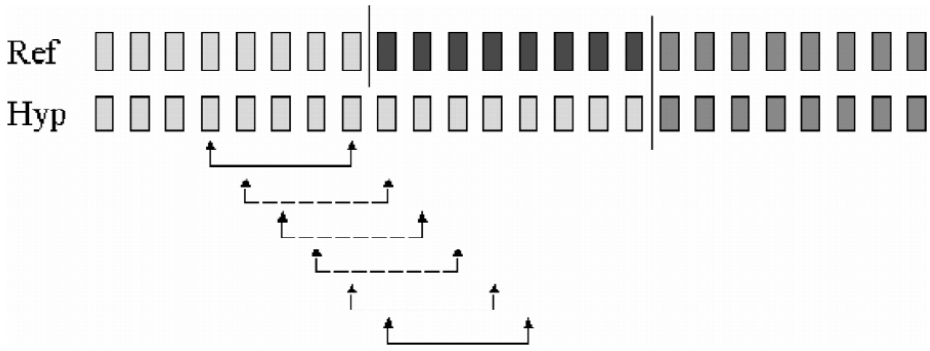

Figure 3.2.: *Demonstration of the weakness of P_k [PH02]*

This happens due to the ignorance of P_k regarding the *number* of topic boundaries occuring between the ends of the sliding window in the proximity of a real boundary. All the windows in the example would not add to the mismatches score because in all cases there is no disagreement between H and R whether there is a boundary between window's ends or not. To trap such situations [PH02] modifies P_k to recognize a match between H and R only if the same number of topic boundaries $b_S(i, j)$ is constituted. So the indicator function is then

$$|b_H(i,j) - b_R(i,j| > 0 \tag{3.7}$$

This means, a mismatch between evaluated (H) and referenced (R) segmentations is given if there is a different number of topic boundaries that can be found inside the sliding window. If the number is the same in both cases, b is 0 and is not taken into account to calculate the *WindowDiff* measure.

$$WD = \frac{\sum_{i=1}^{N-k}[|b_H(i,i+k) - b_R(i,i+k| > 0]}{N-k} \tag{3.8}$$

3.1.4. Pr_{error} metric

Not yet widely accepted, Pr_{error} proposed by [GCA06] bases on the observation that WD scores misses better than false alarms. However, as also pointed out by [MMW09], a miss can lead to mixing up semantically divergent content, which is more critical than creating an unnecessary boundary inside a topically coherent segment. Specifically, WD_{Miss} and $WD_{FalseAlarm}$ can be derived from WD ($WD = WD_{Miss} + WD_{FalseAlarm}$) , which are both normalized by the number of sliding windows, which is the same as the number of maximum possible occurrences for false alarms. However, misses might occur more often than false alarms. To account for that, a normalization of misses to the number of their possible occurrences should be preferred instead (i.e. number of windows containing topic boundaries from the reference segmentation). Therefore, Pr_{error} can be calculated as the following.

$$\begin{aligned} Pr_{error} &= 0.5 \cdot (Pr_{Miss} + Pr_{FalseAlarm}) \\ &= 0.5 \cdot (\frac{\sum_{i=1}^{N-k}[b_H(i,i+k) < b_R(i,i+k]}{\sum_{i=1}^{N-k}[b_R(i,i+k) > 0]} + WD_{FalseAlarm}) \end{aligned} \tag{3.9}$$

The coefficient 0.5 reflects the costs of misses and false alarms. It is chosen due to the consideration that it should be the same for both of them, delivering $Pr_{error} = 50\%$ for a degenerate algorithm (all possible or no boundaries).

3.1.5. Topic closeness measure (TCM)

In [MMW09] it is argued that P_k is dependent on the choice of k and suggests an exact matching of block segments in the hypothethized and reference segmentations. This means that it is hardly possible to compare a text segmentation with its transcribed version. The authors propose to change from a segmentation-based to a

content-based error metric. They elaborate a probabilistic similarity measure K_{norm} based on the *pointwise mutual information* (PMI=$\log \frac{p(x,y)}{p(x)p(y)}$ for variables x, y), which is used in information retrieval as a measure of association. In this context it is used to describe closeness of words. Given reference segmentation R consisting of k segments and hypothethized segmentation H consisting of l segments, TCM is defined as the following.

$$TCM = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l} Q(i,j) K_{norm}(r_i, h_j)}{\sum_{i=1}^{k} \sum_{i=1}^{l} Q(i,j)} \tag{3.10}$$

Q is an indicator function returning value one if segments i and j overlap and zero if not. TCM is reported to be strongly correlating with P_k and is aware not only of the boundary placement but also of the content separated by them.

3.2. Conventional corpora overview

Besides quality metrics algorithms, there is a need in testing material to apply segmentation algorithms to it. The amount, composition character and complexity of data sets can have large impact on the output quality (e.g. due to noisiness or the extent to which the data represents the target domain). In computer linguistics, a data set used for algorithm development and evaluation is referred to as *corpus* (pl. *corpora*), which is the Latin word for "body". A corpus is usually a set of text files (assets) sharing some common criteria (e.g. corpora containing news or meetings transcripts). Table 3.1 gives an overview on corpora used in publications on topic segmentation presented in the next section[1]. Corpora referred to be artificial consist of files created through concatenation of text fragments originating from different sources. More details on designing corpora for topic segmentation research can be found in [Way00] and [LP08].

As stated in [LP08], "artificial corpora could favor techniques sensitive to clean cuts in topics, whereas natural corpora would introduce a higher difficulty, since transitions are smoother, and so topic shifting more difficult to detect." This statement allows the assumption that algorithms developed with artificial corpora are less suitable for hard real-world data than news broadcasts, which exhibit similar clean cuts in topicality. This means that finding algorithms, being able to detect smooth topic transitions might require more sophisticated approaches than building on abrupt topic changes.

[1] Most of these corpora can be found at the Linguistic Data Consortium website, http://ldc.upenn.edu/.

Year	Name	N assets	Description
1967	Brown corpus	500	part-of-speech annotation, about 10^6 words
2000	RCV1	810.000	news stories in English language (Reuters and CNN); 2.5 GB data
2000	Choi's corpus	700	artifical corpus based on the Brown corpus
2001	TDT3	19.000	multiple languages and media; contains ASR output
2004	ICSI	75	ASR meetings transcripts
2005	RCV2	487.000	news stories in 13 languages (Reuters)

Table 3.1.: *Overview of conventional text corpora used for algorithm evaluation referred in publications presented in this work.*

3.3. Classification of known approaches

Although the practical goal of this work is to find well-performing segmentation algorithms for ASR transcripts, there has been less research sticking exclusively to them rather than treating text segmentation in general. Therefore this section summarizes known topic segmentation approaches, pointing out where a method was also tested with an ASR transcript. It is however important to keep in mind that the quality of an ASR system output can have an essential impact on topic segmentation concepts by adding noise to topic-specific word distributions in the transcript. The following classification is an effort to group found publications by applied methods. However, another approach could be e.g. to classify methods by applied text similarity measures. The overview is thought to be as complete as possible to name all discussed methods and their performance, which means providing only general information on the algorithm design. The purpose is to identify best-performing topic segmentation algorithms and give some information on their background. Figure 3.3 on page 22 gives a chronological overview on methods and research publications on topic segmentation since the first publications in the early 1990s.

3.3.1. Common preprocessing steps and general algorithm layout

Most topic segmentation algorithms share common steps and can be generalized on some abstract level. Fig. 3.4 on page 23 shows the layout of an intuitive approach for topic segmentation.

If we aim to determine topic boundaries in a written text, it will be usually normalized by removing punctuation marks (a) and segmented by sentence or paragraph boundaries (b). In the case of ASR transcript segmentation can be done using

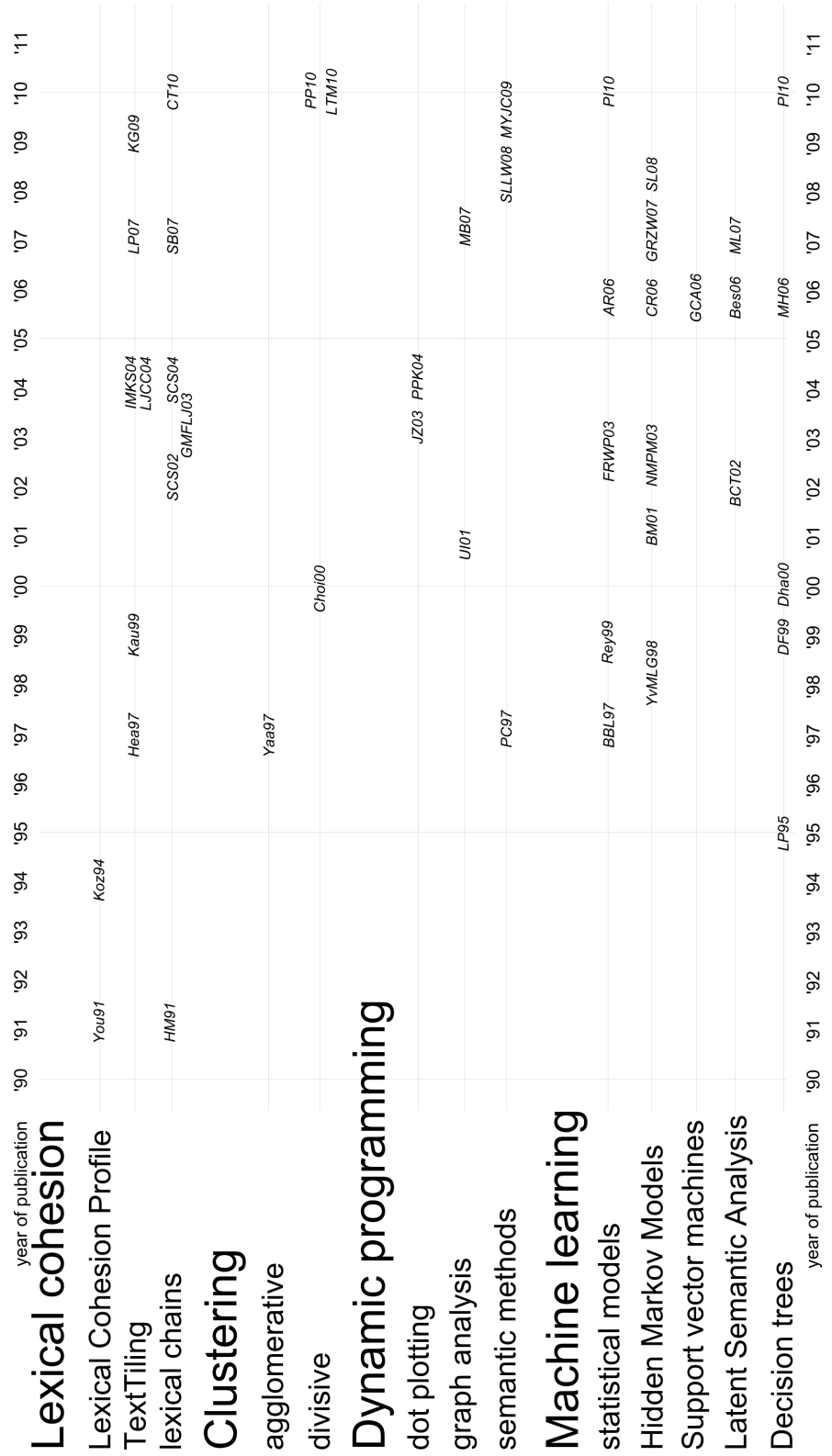

Figure 3.3.: *A chronological overview of publications on topic segmentation and applied methods.*

non-speech or silence fragments, which exceed some threshold by their duration. If this data is not present, a simple uniform tiling can be also applied (c). This pre-segmentation step is done to establish comparable segments and candidate places where further processing would place topic boundaries. However, not all approaches need this, since topic boundaries can be naturally placed after virtually every word (with exception of violating sentence boundaries, if applicable). An advanced approach would be to identify sentence boundaries in an ASR transcript [KL10]. In both cases further normalization is usually done by removing stop words and stemming them to their uninflected form or stem.

The intuitive notion of topic segmentation is that topic segments have to be coherent inside them regarding word usage. So if we compare two subsequent text pieces, they will be more similar if they share the same topic than in the opposite case (e). Based on this notion, a *similarity curve* can be computed by comparing all subsequent text fragments of text to each other. This can be also done in two dimensions by plotting a similarity matrix of the whole document. The idea is that different topics would have their boundaries where the similarity curve has its minima, or respectively between the most dense regions exhibited by the similarity matrix (f). Similarity of two text pieces is very often computed as the *cosine measure* of their vectorial presentations, based on word frequencies (orthogonal vectors correspond to the highest possible extent of dissimilarity resulting in zero dot product, or no similarity).

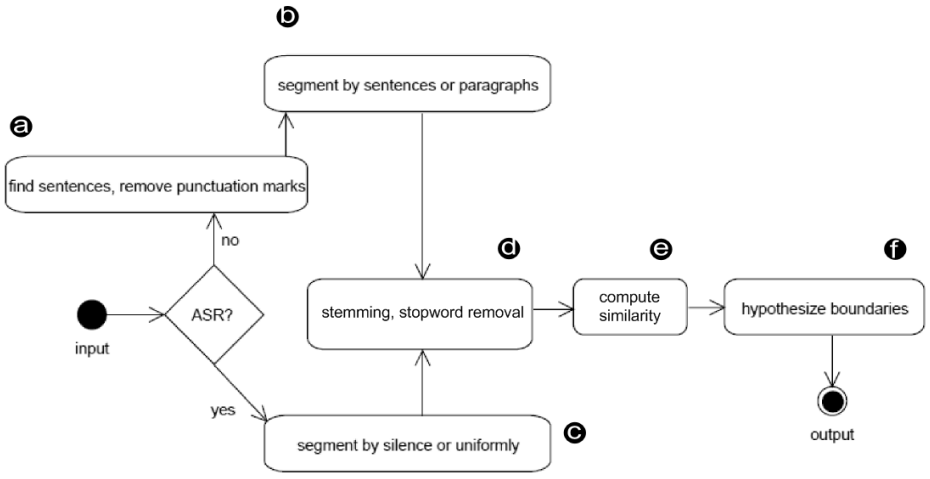

Figure 3.4.: *Intuitive layout of an unsupervised topic segmentation algorithm.*

More advanced methods consider topic segmentation to be a *classification problem*. This means that an algortihm previously trained on sample data would make a decision for every given segment boundary if it is a topic boundary or not. For more details please consider Fig. 3.12 on page 37.

3.3.2. Methods treating lexical cohesion/lexical similarity

Most of the earliest publications on topic segmentation are widely influenced by the theoretical foundations by [HH76] and the observation that topically coherent text segments employ the linguistic phenomenon of lexical cohesion (see page 2.1). This subsection gives an overview of approaches elaborated in this research direction.

Lexical cohesion profile (LCP)

An early method proposed by [You91] suggests finding topic boundaries by creating a vocabulary management profile (VMP) through identification regions containing peaks of firstly used words. VMP is used to determine topic boundaries in [NN94] in connection with Japanese topic-demarkating part of speech *wa*(F-score of 66%). [Koz93; KF94] argues however that a text with high terminological density does not have enough word reiterations (e.g. topically equivalent segments with *no* new words).

He suggests generating a *lexical cohesion profile* of a text through determining a sequence of lexical cohesion values for each word given by a sliding window of fixed width employing a large and already available semantic network (Figure 3.5) of English language. Each word list given through a position of the sliding window

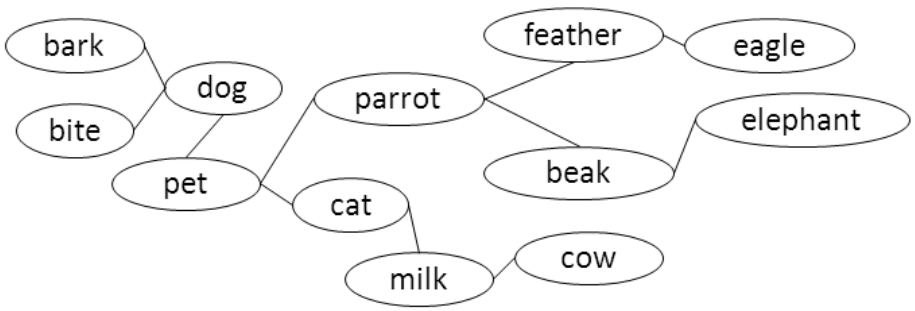

Figure 3.5.: *Example of a simple semantic network: a semantic network is a form of knowledge representation. It can be visualized as a graph, nodes of which represent terms. The edges represent associations between the terms.*

is used to compute its cohesiveness as a sum of activation values in the semantic network for each word (*spreading activation*). For example, a word list *Molly saw a cat it was her family pet she wished to keep a lion* results in a higher cohesiveness value than *there is no one but me put on your clothes I can not walk more.*

Figure 3.6 shows a good correlation between LCP minima and corresponding human judgements histogram. However, apparently no evaluations on big corpora neither usage of error metrics were reported.

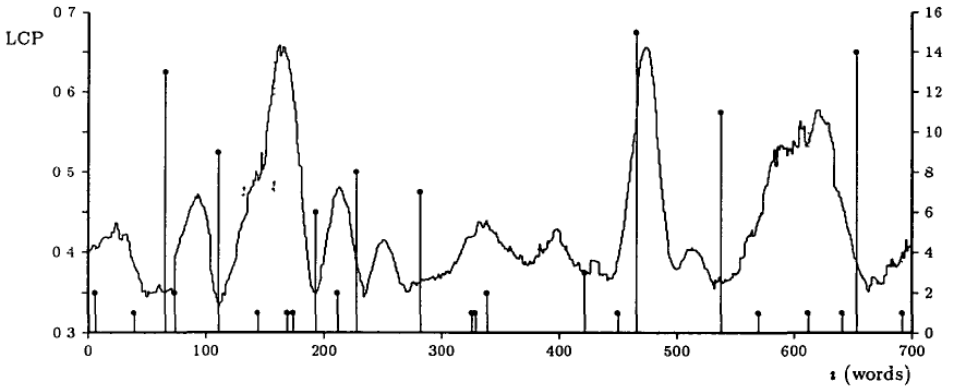

Figure 3.6.: *Example of an LCP graph; graph minima correspond to topic boundaries with least lexical cohesion for a sliding window at this point. Vertical solid lines are topic boundaries proposed by human annotators (higher values correspond to more votes given by a test group of 16 people). [Koz93]*

An interesting feature of LCP is that it is able to work on a normalized text stream containing no logical structure, which makes it interesting for ASR transcript processsing. However, a large semantic network is required and a similarity measure based on it depends to a considerable extent on its quality and up-to-dateness especially regarding news broadcasts. The threshold used for a local minimum to qualify as a topic boundary is also ambiguous.

TextTiling and derivatives

One of the most influential topic segmentation algorithms is TextTiling [Hea97]. It bases on the assumption that two adjacent text segments would have overlapping words if they are topically coherent. TextTiling applies uses a sliding window over k sentences and computes their pairwise similarity. The value of k is suggested to be an average paragraph length in sentences. A dynamic programming technique can be used to preprocess block boundaries involving lexical similarity and cost functions of grouping sentences as shown in [Hei98]. A cosine metric based on the *tf-idf* tf-idf metric is then applied to compute the word list, or block similarity employing the sliding window technique:

$$sim(b1, b2) = cos(b1, b2) = \frac{\sum_{i=1}^{n} w_{i,b1} w_{i,b2}}{\sqrt{\sum_{i=1}^{n} w_{i,b1}^2 w_{i,b2}^2}} \tag{3.11}$$

n is the number of all words in the asset and $w_{i,bx}$ ($x \in \{1, 2\}$) is the *tf-idf* weight assigned to a word i in its containing block. *tf-idf* is a common information retrieval metric which denotes in this context a word's importance per window. Finally, the number of small local minima is reduced by a simple smoothing algorithm. Figure 3.7 shows an example segmentation produced by TextTiling. Topic boundaries are then hypothethized by means of a *depth score*, a sharpness metric of a graph minimum m: $(sim_{lmax} - sim_m) + (sim_{rmax} - sim_m)$, where $lmax, rmax$ stand for the left and the right peaks surrounding m.

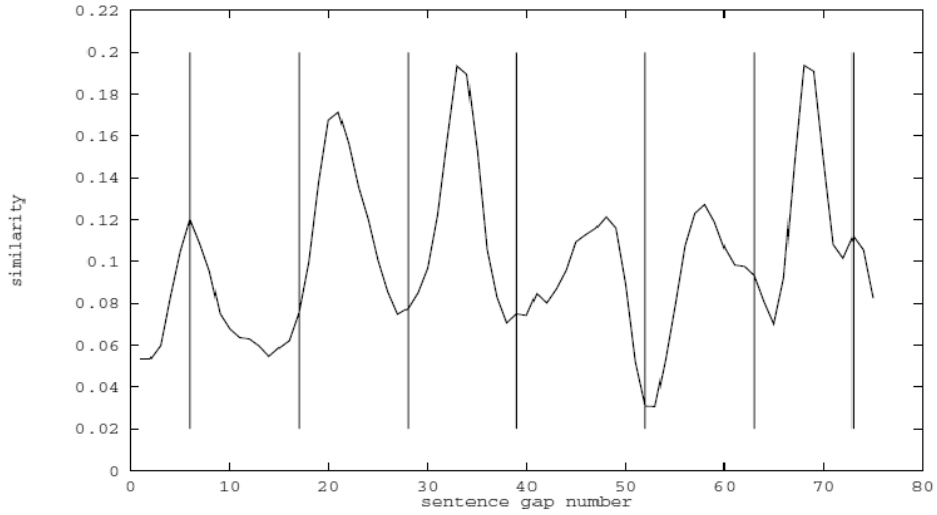

Figure 3.7.: *Example of a similarity curve produced by TextTiling; graph minima correspond to topic boundaries. Vertical solid lines are real topic boundaries proposed by human annotators. [Hea97]*

[Hea97] reports precision and recall to vary depending on assets and real topic boundaries number between 80% and 30% and 30% and 92% respectively. An important performance feature of TextTiling is its linear ($\mathcal{O}(n)$) runtime complexity.

[Ful08] tested TextTiling on a data set of 30 podcast episodes. The algorithm shows similar performance on both manual and ASR transcripts (1% lower precision and 4% higher recall for ASR output).

[Kau99] points out that TextTiling considers only exact matches of words and employs a dictionary-based similarity measure to determine semantic similarity. He reports an average improvement of 20% for recall and 11% for precision. However, different test sets were used (newspaper articles on popular science). This specific corpus disallows an adequate comparison.

A modification is proposed by [LJCC04] to use more features for the computation of the similarity measure. Their similarity measure employs seven feature vectors of two kinds. They define five content-based features (noun phrases, verb classes, word

stems, topic words, combined features) and two discourse-based (pronouns and cue phrases). The similarity measure for two adjacent window positions b_1, b_2 is then defined as the following:

$$sim(b1, b2) = \sum_j \frac{\sum_i f_{i,j,b_1} f_{i,j,b_2}}{\sqrt{\sum_i f_{j,i,b_1}^2 \sum_i f_{j,i,b_2}^2}} S_j \tag{3.12}$$

In this case the similarity function is calculated between feature vectors (e.g. f_{i,j,b_1}) instead of word frequencies. The paper employs also a specific importance measure instead of *tf-idf* as well as the general feature frequency S_j based on a language model. The evaluation was done on three manual lecture video transcripts resulting in 77% precision and 67% recall if allowed "fuzzy matching" of boundaries (e.g. up to one sentence away from the actual topic boundary). A similar approach is used by [IMKS04] with more high-level feature vectors built upon four semantic classes (general, personal, locational/organizational, or temporal).

In [LP07] part-of-speech annotation and a natural language parser are used. Sliding window is considered to be consisting of two potential segments of equal length corresponding to the two centroids of contained sentences. The value of the similarity curve for each window position is then the thematic distance between the centroids (angular distance measure). Topic boundaries are found through thresholding. This technique results in 16.4% precision and 80% recall which means the method produces an oversegmented output.

A further improvement is to calculate also similarities inside the blocks and then pairwise [KG09](i.e. building context vectors one per sentence in a block and not one per block). This new approach, called TSF, is reported to deliver a comparatively very low error rate P_k=5.3%. Evaluation was done on large corpora like Reuters corpus RCV1 consisting of 810.000 news stories; another corpus, RCV2, included stories in 13 different languages.

Lexical chains

The term *lexical chains* is elaborately discussed in [HM91]. The authors constitute that words spanning a topical unit in a text form cohesive binds also if they are distant and that lexical cohesion can be found beyond simple word reiteration. Lexical chains are then sequences of related or repeated words (Fig. 3.8).

[SCS02; SCS04] point out that lexical chains are formed by semantically clustered terms. Possible lexicographical relationships between them are explained with the following example:

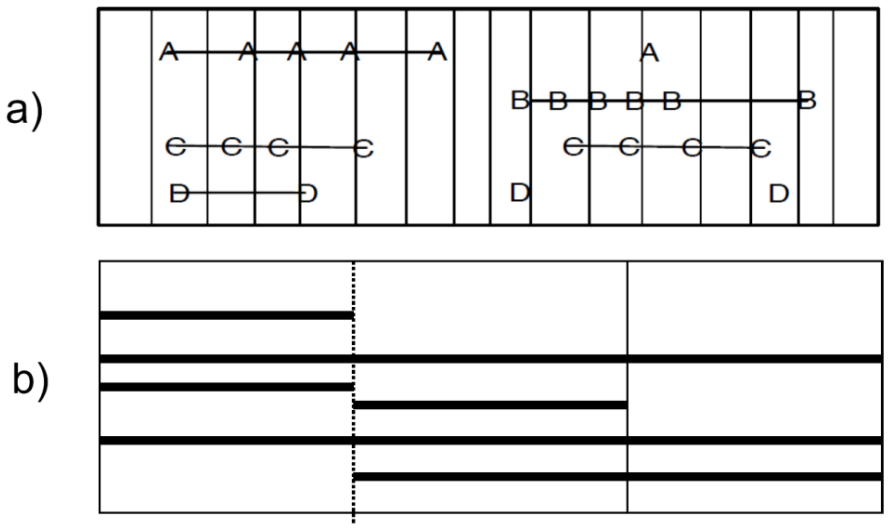

Figure 3.8.: *a) Lexical chains spanning sequences of lemmas A, B, C and D [SB07]. b) Abstract representation of lexical chains; the dotted line demarkates a hypothethized topic boundary as the position coinciding with the most chains starts and ends [SCS04].*

"For example in a document concerning cars a typical chain might consist of the following words {*BMW, vehicle, engine, wheel, car, automobile, tire*}, where each word in the chain is directly or indirectly related to another word by a semantic relationship such as *synonymy* (car and automobile are semantically equivalent), *holonymy* (car has-part engine), *hyponymy* (BMW is a specialization of a car), *meronymy* (tire is part-of a wheel) and *hypernymy* (vehicle is a generalisation of a car)" [SCS02].

A boundary between topics can be then hypothethized at places where high concentrations of chain start and end points occur. The segmentation algorithm SeLeCT described in [SCS04] is reported to have an error rate P_k of 25%. Chains are created in a single-pass clustering procedure which tries to add a word to an existing chain by evaluating relationships involving a large external thesaurus. Boundary detection between segments n and $n+1$ is then computed based on the *boundary strength* which is the sum of adjacent chain end and start points counts: $s(n, n+1) = |n_{endpoints}| + |(n+1)_{startpoints}|$.

A more influential algorithm LCseg by [GMFLJ03] determines topic boundaries in a similar way, looking however only for chains consisting of simple word repetitions. Weakly linked chains are avoided through breaking up longer chains containing large gaps while the threshold is experimentally determined. The core algorithm bases on weighting identified chains and finding topic boundaries correlated with boundaries of multiple parallel chains, referring to the same topic segment. Weighting is computed based on the word frequency and the compactness of a

chain: $score(C(t)_i) = freq(t_i) \cdot log(\frac{T}{|C(t)_i|})$ where C_i is a chain of term t and T the text length (all length counts are in sentences; note that a chain contains all document words spanned by a term iteration). In a second step a sliding window technique is employed to compute lexical cohesion between chains at each sentence break or which might be more interesting for an ASR transcript, at each speaker turn (however, this is correct only for multiparty dialogues; [FRWP03] reports this feature to have minor influence). The similarity measure for all chains C is again the cosine metric for a pair of adjacent windows $b1$ and $b2$, where $\gamma \in \{b1, b2\}$ and $w(C)_{i,\gamma} = score(C)$ if C overlaps γ, otherwise $w(C)_{i,\gamma} = 0$. So, the cosine metric is defined as the following:

$$sim(b1, b2) = \frac{\sum_i w_{i,b_1} \cdot w_{i,b_2}}{\sqrt{\sum_i w_{i,b_1}^2 \cdot \sum_i w_{i,b_2}^2}} \tag{3.13}$$

After applying a smoothing filter (moving average), for each local minimum a boundary probability $p(m)$ is computed by determining the sharpness of a valley similarly to depth scores proposed by [Hea97] m:

$$p(m) = \frac{1}{2}[sim_{lmax} + sim_{rmax} - 2 \cdot sim(m)] \tag{3.14}$$

Final selection of detected topic boundaries is done by a two-step thresholding. All local minima below a manually set p_{limit} are omitted and the second threshold is defined as $\mu - \sigma$ (difference of average and standard deviation). Depending on used corpus, a P_k error rate of down to 6.95% (TDT) is reported. However, [GCA06; SL08] report it to have P_k=32-35% on the ICSI corpus.

An improvement is proposed by [SB07] to weight every term repetition through its chain (weighted lexical links, WLL). This leads to a slight advance in performance, WD_{WLL}=31.87% over WD_{LCSeg}=32.72% error rate in their setup.

More advanced concepts were used in [CT10] combining lexical chaining with work of [Kat96] to identify decision-making in transcripts. They report P_k=23% which is constituted to be better than TextTiling.

3.3.3. Clustering methods

Previously described methods deal with identifying areas of low cohesion (e.g. topic boundaries). An alternative approach is to look for high cohesion areas (e.g. the topics segments).

Two ways of clustering data are generally used, *hierarchical agglomerative* and *divisive* clustering.

Agglomerative clustering

In an agglomerative clustering algorithm all data samples (e.g. words or sentences are initially assumed to be single clusters). After that the two most similar clusters are merged into one new cluster (bottom-up approach). The algorithm terminates after determining a preset number of clusters or after merging all clusters into one single cluster. In the latter case a binary tree, or *dendrogram* of the data set is created. However, possible clustering errors propagate towards the tree root which means potential inaccuracies on the top level (i.e. the topic segmentation).

A method based on agglomerative clustering was proposed by [Yaa97]. The dendrogram is then used to calculate depth profile of paragraphs similar to depth scores in [Hea97]. The basic assumption here is that nodes which are close to roots should be representative for topics. Experiments conducted on the same data as [Hea97] deliver good results: 87% precision and 78% recall.

Divisive clustering

Divisive clustering is a top-down approach. All data samples are considered to be one single cluster. Every existing cluster is then split into two maximally dissimilar clusters.

[Rey98] developed a method based on clustering a two-dimensional matrix expressing text similarity generated by the *dot-plotting* technique. A dotplot chart is a binary matrix based on word comparison w_i and w_j where $S_{i,j} = 1$ if $w_i = w_j$ and $S_{i,j} = 0$ otherwise (Fig. 3.9). The diagonal line denotates the self-similarity of the text; more interesting are good recognizable darker square areas resulting from higher dot density. These square areas can be considered to be topics because they represent frequently occuring words in distinct text segments. Formally, clustering is done by determining the boundary set for which the outside density of resulting squares is minimal. However, this approach assumes that the number of topics is known, which is the general problem of divisive clustering. This drawback of the dotplotting method can be overcome by applying dynamic programming for segmentation instead of clustering [JZ03; PPK04](see 3.3.4).

Very influential findings were presented by [Cho00], building upon the initial dotplotting approach; double accuracy and maximum increase of 600% in execution speed compared to [Rey98] were reported. The algorithm called C99 bases on the

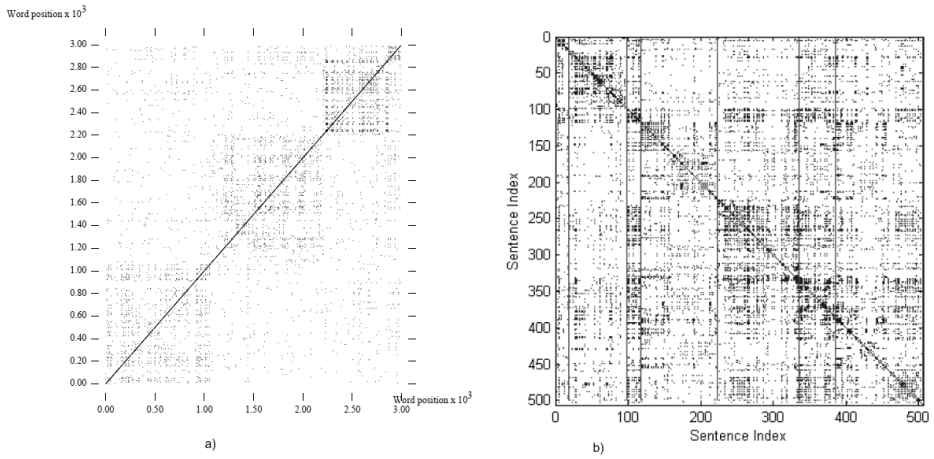

Figure 3.9.: *Dotplotting topics: a) four journal articles [Rey98]; b) ASR output for a lecture, vertical lines demarcate real topic boundaries [MB07].*

assumption that similarity of short text fragments is less important than of bigger ones and introduces a ranking scheme for them.

The similarity matrix is then calculated using the cosine measure for normalized sentence pairs ($f_{i,j}$ is the frequency of word j in sentence i):

$$sim(x,y) = \frac{\sum_j f_{x,j} \times f_{y,i}}{\sqrt{\sum_j f_{x,j}^2 \times \sum_j f_{y,j}^2}} \tag{3.15}$$

The author argues that absolute values of $sim(x,y)$ are not reliable as they may be disproportional for short segments and furthermore, different text segments cannot be directly compared due to variating cohesiveness. Therefore a ranking is proposed. A rank of an element corresponds to the number of neighbouring elements with lower similarity value. After that, divisive clustering is applied based on [Rey98]. The algorithm terminates automatically based on an experimentally discovered outside density threshold value.

C99 is reported to have the error rate P_k of 12% compared to 22% for [Rey98]. While C99 is widely referenced, some recent publications (e.g. [GCA06]) evaluating it with less artificial corpora than in [Cho00] report poor performance compared to other approaches. Thus, C99 shows P_k=21% on the TDT corpus and P_k=55% on the ICSI corpus, both corpora being speech transcripts. [MB07] shows C99 to perform with P_k =35.2% on ASR transcripts of spoken lectures.

A novel approach is proposed by [LTM10] based on the *formal concept analysis*. In the first step, an ontology is derived from the text where transitive verbs build objects and attributes are corresponding nouns. This information is then used to cluster

sentences based on distancies of concept vectors applying $k-means$ clustering but on concept level. Due to this fact, many segmentations can be derived simultaneously providing concept-oriented *views* on the text. For example, for a text consisting of 30 sentences, one cluster might contain sentences $\{S_4, S_{10}, S_{22}\}$ and the other one sentences $\{S_{12}, S_{18}, S_{23}\}$. Thus, corresponding segmentations are $\{[S_0 : S_3], [S_4 : S_9], [S_{11} : S_{21}], [S_{22} : S_{29}]\}$ and $\{[S_0 : S11], [S_{12} : S_{17}], [S_{18} : S_{22}], [S_{23} : S29]\}$ (i.e. each cluster element demarkates a segment start in a segmentation specific to the concept set of the cluster). No large tests nor performance comparisons to other methods are reported, limiting the evaluation to a single text from the law domain consisting of 270 sentences.

Another novel approach to topic segmentation by means of divisive clustering is [PP10], where the authors propose to assign more than one topic to a paragraph, which can be analysed by an incremental overlapped clustering algorithm. They report an error rate WD=10%.

3.3.4. Dynamic programming

Dynamic programming is a method for solving optimization problems by solving their parts. It is applicable if a problem consists of overlapping (reusable) subproblems which if solved allow construction of an optimal solution to the whole problem. Generally speaking, optimization means finding a solution which is better than all its alternatives by some criteria.

Algorithms based on the dynamic programming are usually not naïve, which can involve problem solving in some other domain. This subsection gives an overview of topic segmentation approaches modeled as optimization problems.

Dot-plotting

[JZ03] uses an extended dot-plotted document model [Rey98]. Instead of binary matrix for matching words, Euclidean distances between sentence vectors are plotted resulting in a greyscale image. Black pixels denote zero distance or equality. Instead of sentences, also other small text units can be used. After that, an image processing technique called *anisotropic diffusion* is applied (Fig. 3.10[2]).

In essence, the image is modified in a way that homogeneous (i.e. topically coherent) regions are consolidated and sharpened. Topic segmentation is then equivalent to the

[2]Note that the images render numeric data which might lead to a subjective impression of "bad" image quality of the hard copy.

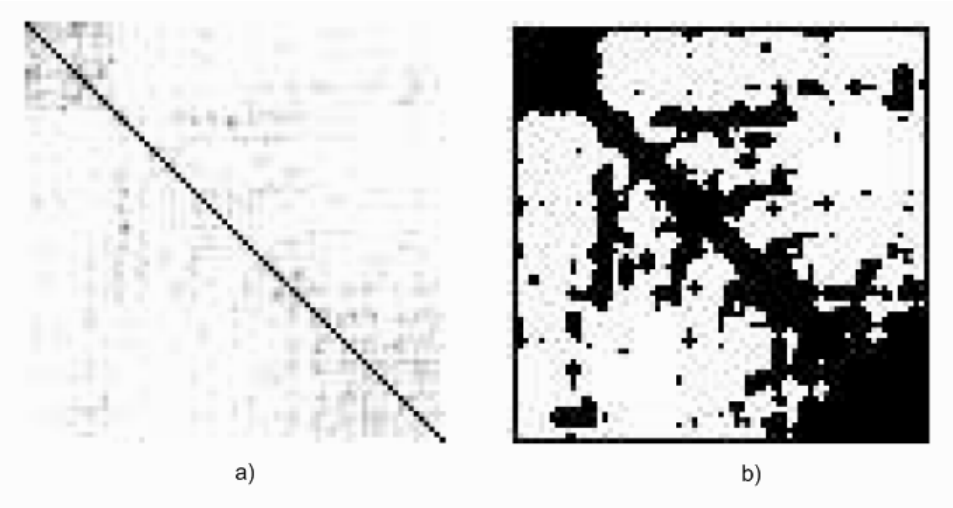

Figure 3.10.: *Sentence distance matrix: a) before, b) after applying anisotropic diffusion [JZ03].*

partitioning of the image into blocks so that an error function has its global minimum where the optimal segmentation correlates to the dark areas.

The algorithm has been tested with Choi's corpus [Cho00] resulting in a comparatively very low error rate P_k=4.3%.

A variation of the previous method is proposed by [PPK04]. In this case a binary sentence similarity matrix is used (1 for at least one common word, 0 for no common words for every sentence pair). Dynamic programming is done by minimizing a cost function which incorporates cost functions of segment length and density. In other words, deviation from preset expected average segment length as well as sparse dot density are penalized. The algorithm is reported to have an error rate P_k under 5% on Choi's corpus. However, this is possible only by determining proper parameters through experiments, i.e., by training on a part of the corpus. Otherwise, the error rate can, as shown, increase up to 25%-45% in the worst case.

Graph analysis

[UI01] models segmentation of a text in terms of the probability theory. Given a text as a word list $W = w_1 w_2 ... w_n, |W| = N$ and its segmentation $S = S_1 S_2 ... S_m$, probability of a segmentation S is

$$P(S|W) = \frac{P(W|S) \cdot P(S)}{P(W)} \tag{3.16}$$

$P(W|S)$ can be decomposed as shown in 3.17, $f_i(w_i^j)$ is the number of words in a segment W_i which are the same as w_i^j and k is the number of different words in W. The decomposition is based on the assumption that different topics are statistically independent of each other and possess different topic distributions; also words appearing in a topic are considered to be statistically independent of each other.

$$P(W|S) = \prod_{i=1}^{m} \prod_{j=1}^{n_i} \frac{f_i(w_i^j) + 1}{n_i + k} \tag{3.17}$$

$P(S)$ is then defined as $P(S) = 2^{-m \log n}$ based on the idea of the *description length* of a segmentation. (Please see [UI01] for more details).

With this model setup, a cost function C for a segmentation S is defined: $C(S) = -\log P(W|S)P(S)$. The maximum-probability segmentation $\hat{S}$ is then minimized $C(S)$: $\hat{S} = \arg\max_S P(W|S)P(S) = \arg\min_S C(S)$.

This allows for problem solving in the domain of graph analysis: a graph spanning the text, one edge per word can be defined (Figure 3.11). The minimum cost path is a well-known problem in graph analysis domain and can be solved using Dijkstra's algorithm [Dij59] for finding the shortest path in a graph.

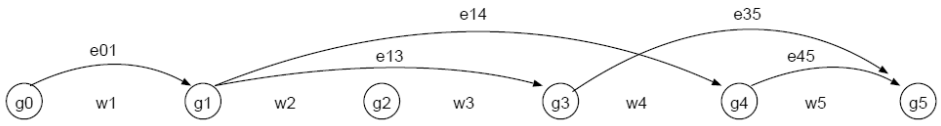

Figure 3.11.: *A graph spanning a text: nodes g_i are placed between words w_j[UI01].*

The method is reported to have an error rate of P_k=10% compared to P_k=13% of C99 evaluated on the artificial corpus from [Cho00]. Evaluation on ASR transcripts shows however worse performance P_k=35.2% [MB07]. The statistical model established by [UI01] has been extended with semantic relations and probability of a word to be recognized by an ASR system [GGS10], however instead of P_k measure the authors disclose their method's performance to have an F-score of 60.8%.

Another graph-theoretical approach is proposed by [MB07]. The authors point out that previous approaches do not consider long-range cohesion dependencies due to the fact that "homogeneity of a segment is determined not only by the similarity of its words, but also by their relation to words in other segments of the text".

A text can be modeled as a weighted undirected graph. The nodes of the graph stand for sentences and the edges get assigned their pairwise similarities as weights. Text segmentation is then handled as graph partitioning optimization problem based on

the *normalized-cut* criterion adopted from the image analysis domain. This metric is used to measure segment homogeneity as well as dissimilarity simultaneously. The best segmentation is then a set of mostly homogeneous segments which are also maximally dissimilar from each other. Pairwise similarity of sentence vectors is computed as exponential cosine similarity to avoid number precision issues with small scores:

$$sim(s_i, s_j) = e^{\frac{s_i \cdot s_j}{s_i \times s_j}} \tag{3.18}$$

The ASR transcripts were simply cut in blocks of fixed length k, determining the best value for k through experiments.

[MB07] reports performance of P_k=32.2% compared to P_k=36.1% for [Cho00] on ASR transcripts. Tests on the corpus of [Cho00] prove C99 to perform approximately twice better which is explained through lacking wide-range cohesion in an artificial corpus which is essential for good performance of the presented method.

Semantic methods

The term "semantic" can be applied to the following methods only due to the fact that they induce *semantic proximities* of words by extracting them from their contexts. No semantic modeling is involved.

The method presented by [PC97] is based on a *query expansion* concept called *local context analysis* (LCA). In essence, this technique allows creation of a concept database searchable for locally associated terms. Each sentence of the investigated text is then queried against the database. Top M concepts (e.g. 100) are then extracted from N top search results (e.g. 2000) and ranked depending on their cooccurrence with the query terms. In this way, every sentence is replaced through its related concepts list. The pairwise similarity measure is then defined by the number of matching LCA features for each sentence pair. Finally, all possible segmentations are scored through summarizing the similarity measure for each block in question. Though there are exponentially many possible segmentations, the dynamic programming approach allows acceptable runtime. The authors do not reveal further details on the implementation. The runtime complexity is mentioned to be $\mathcal{O}(n)$. The quality of segmentation is measured to be 82.6% for precision and 88.8% for recall.

[SLLW08] employ a more complex model based on the concept of *latent Dirichlet allocation* (LDA), building on a set of sophisticated statistical analysis methods. The general idea of LDA is that a document can be modeled as a mixture of latent

topics resulting from a generative process. Their distribution is assumed to have the Dirichlet *prior probability distribution* .

In [SLLW08], segment similarity is then measured by the *Fisher kernel*, which shows if two adjacent segments adopt the LDA model in the same way based on the underlying probability distributions (e.g. to which extent they share the same latent topics). The method involves creation of a corpus vocabulary. The dynamic programming cost function minimizes the Fisher kernel output, considering also the segment length as in [PPK04]. Tests on an artificial Chinese news corpus deliver the best result $P_k \leq 5\%$ given segment lengths in a range of 13-15 sentences.

[MYJC09] proposed an LDA-based topic segmentation approach combined with dynamic programming and graph analysis. Two following experiments were conducted. The system was trained on a news corpus and tested on the Choi's corpus (E1). Then, another instance was trained on a hybrid corpus consisting of the news corpus and a part of Choi's corpus, and tested on another part of Choi's corpus. The results are $P_k(E1)$=23% and $P_k(E2)$=2.2% for block widths of 3-5 sentences. This performance difference of almost one order of magnitude can be explained through the vocabulary sensitivity of LDA.

Other methods

Similarly to the normalized cut criterion [MB07] or anisotropic diffusion [JZ03], in [YZZ+08] a concept is proposed to find mostly topically self-coherent and dissimilar segments. The presented method is based on the idea to find a segmentation scoring function which would increase if both segment dissimilarity and the self-coherence increase. However, with P_k=37% this approach delivers no improvements of known methods. This method also requires parameter training.

3.3.5. Machine learning

A good overview of the research field dealing with machine learning is given by [Mit06]. More intense disquisition can be found in [Alp10]. Machine learning originates from the intersection of statistics and computer science and, generally speaking, investigates adaptive methods of exploring data. "... A machine learns with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E" [Mit06].

Most commonly used are *supervised* and *unsupervised* methods. In supervised learning a function has to be inferred from training data which maps inputs to outputs

in a desired way. In the training data set, pairs of such mappings are provided. In unsupervised learning it is required to derive a model explaining data organisation.

From this point of view, some dynamic programming methods mentioned above which used training are already examples of supervised machine learning [PPK04; YZZ+08], as well as any other method involving the optimal parameter tuning by means of training data set.

In terms of machine learning, topic segmentation can be seen as a binary classification problem (i.e. labeling boundaries of text segments as being topic boundaries or not).

The following subsection presents publications employing machine learning approaches; due to the existing thesis limitations and the large set of existing publications a very general overview is given.

General algorithm layout

All supervised methods share common steps and can be generalized on some abstract level. Fig. 3.12 shows the layout of this approach for topic segmentation.

As previously described (see p. 23), input data will be usually normalized before further processing (a-d). Then, the data corpus is bisected to the training and test data sets (e). Through the training phase a learning algorithm learns parameteres needed to adjust it for most perfect matching of reference segmentation provided as additional input. After that, a classification of segment boundaries is done (f).

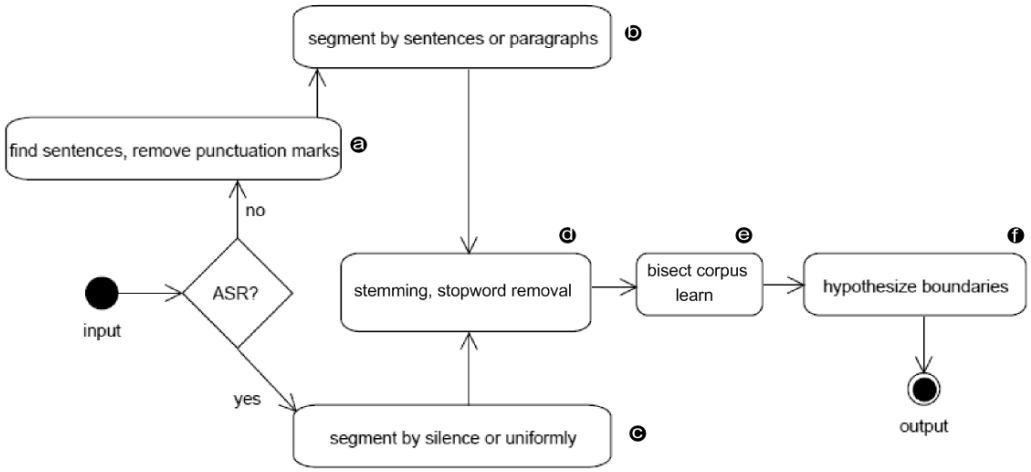

Figure 3.12.: *General layout of a supervised topic segmentation algorithm.*

Statistical models

As described in [SM08], a statistical topic model assumes a text stream to be a stochastic (random) process , generating topics as probability distributions over words based on *latent variables* .

The approach described in [BBL97] is based in general on the idea of using two probability models of different ranges to predict words and to compare their performance (*relevance feature* measuring topicality of context). One of the models is trigram-based and performs in a short range (predict next word based on two previous ones). The second model performs over a longer range (e.g. 20 sentences backwards (history) and N=500 words forth). It predicts words based on probabilities of corpus-specific word pairs. For example, finding the word *flower* increases the probability of finding the word *petals* in the next N words. Experiments show that there is a difference in the performance between the models specifically at places where topic boundaries occur. This can be explained by the fact that the long-range model is influenced by its history and, if the topic has changed, it is no longer "up-to-date" and the history needs to be refilled before the predictions become relevant again. This can be seen in the plot in the Figure 3.13 which shows the log ratio of both models' output around a topic boundary. Additionally, automatically induced vocabulary features were applied to compute potential boundary probabilities (final decisioning was made by a threshold).

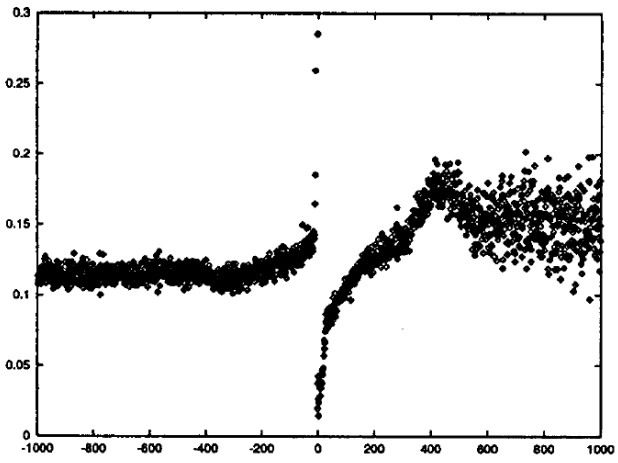

Figure 3.13.: *The average of the log ratio of the adaptive language (long-range) model to the static trigram model* $R(H,w) = \log \frac{p_{exp}(w|H)}{p_{tri(w|w-2w-1)}}$ *[BBL97]. The long-range model tends to produce much lower probabilities from the beginning of a new segment (zero position) which corresponds to the graph fall.*

This publication has also originally proposed the P_k error metric and reports P_k=12%-18% for ASR news transcripts using the presented method.

[Rey99; FRWP03] use models based on *maximum entropy* regarding distributions of part-of-speech or n-grams. These methods report precision/recall of 59%/60% and P_k of ≈45% respectively.

Finally, [PI10] employed a *Bayesian network* combined with lexical chains as features resulting in recall of 79% and precision of 84%. A Bayesian network is a graph analysis approach to design a probabilistic model. It is a *directed acyclic graph* (DAG, a directed graph having no directed cycles) with nodes representing unknown variables. An edge between two nodes visualizes a conditional dependency of their corresponding variables. If two nodes are not connected, this means they are conditionally independent. In terms of topic segmentation, dynamic Bayesian networks can be used to model a sequence of topics. A combination of a Bayesian network based on classifying cue phrases together with evaluating lexical cohesion showed an improvent of P_k down to 10.5%, however, the method is corpus-biased [AR06].

Hidden Markov Models

More complex than assuming just one stochastic process underneath a topic model is using the *Hidden Markov Model* (HMM). A HMM consists of two processes. The first one (*Markov chain*) is not observable (hidden) and undergoes state changes based on an unknown stochastic transition function. What can be seen are only output tokens generated by the second process. This output has its own probability distribution which depends on the states of the hidden process.

The authors in [YvMLG98] point out that the HMM approach is applicable to the task of topic segmentation, if we assume topic transitions to be states of the hidden process. The words or sentences are, in this case, the observable output. It is also stated that there is a certain similarity to the speech recognition task, where hidden phoneme state transitions result in word utterances.

The hidden states, or topics, were modeled through the *k-means* clustering of the corpora, delivering word distributions for each emerging topic. It was then proceeded with an adaptation of a speech recognizer software. This method delivered good results on ASR transcripts with a P_k-like scoring, recall of 81.9% and precision of 80.5%. [NMPM03] reports for it P_k=18.2 on ASR transcripts of about 4000 dictated medical reports. [SL08] provides an investigation on how the number of hidden states and removing stop words can influence the HMM approach (e.g. removing stop words results in a light decrease of the error rate).

[BM01] extends this concept with the PLSA model, introducing the AHMM (aspect HMM) concept. As shown in Figure 3.14, it outperforms the HMM method for

window widths under 200 words (due to the approximation scheme used), having about $P_k \approx 40\%$ on average.

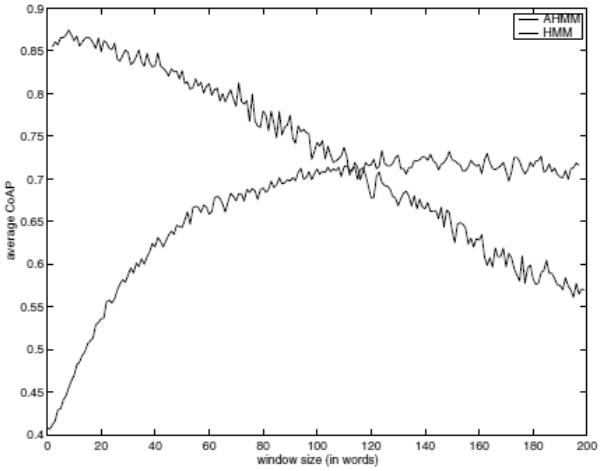

Figure 3.14.: *Aspect HMM performs better than HMM for window sizes under 200 words. The y-axis displays the error metric CoAP=1-P_k. [BM01]*

Another HMM variation is proposed by [NMPM03], extending the topic model through describing the lengths and allocations of sections occupied by the topic segments. This approach shows good performance with P_k=8.5%, however on a very specific ASR corpus consisting of medical reports.

In [GRZW07] (HTMM, *hidden topic Markov model*) it is assumed that topic transitions can happen only between sentences, which emphasizes the fact that consequent words tend to be on the same topic. [MMW10] tested this approach on the TDT corpus and reported P_k=33.1%.

Finally, in [CR06] a *forgetful* HMM is introduced, e.g. there is a topic-neutral state which is passed each time on topic change. However, no evaluation based on an error metric is reported. An interesting feature is that topic boundaries can be placed after each word.

Support Vector Machines

Support vector machines (SVM) are a sophisticated and powerful approach to data classification. Consider the binary linear classification problem shown in Figure 3.15(a). Both classes can be separated through a line or a *hyperplane* – in this case a normal plane. The SVM approach identifies it by maximizing the space between the hyperplane and the closest objects. This assures the correct classification of new objects which are more similar to each other than the objects in the training set. The learning aspect is here to learn the *support vectors* expressing the optimal

(maximum) distance between the closest objects and the separating hyperplane with the help of *kernel functions*.

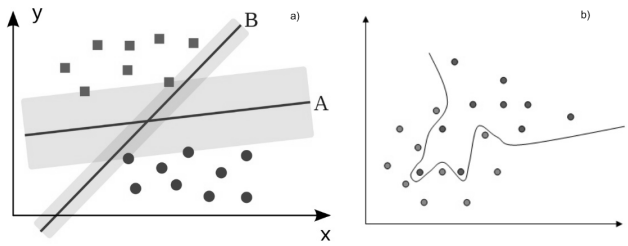

Figure 3.15.: *Linear vs. non-linear classification. [Bü10; Ste10]*

However, real-world classification problems can often be non-linear (Figure 3.15(b)). SVM intervenes here by means of finding the hyperplane in a higher-dimensional space than the problem space. A comprehensive introduction to support vector machines can be found in [CST10].

[GCA06] apply the SVM approach to the task of topic segmentation, calculating SVM input as a vector set computed from a word and its context determined through a sliding window. Tests on the ICSI data set delivered P_k=21.68% and on the TDT data set P_k=20.49%. [MMW10] report similar results on the TDT corpus with P_k=24.4%.

Latent Semantic Analysis

Latent semantic analysis (LSA) is a patented natural language processing technique, finding a large variety of applications in information retrieval.

The main idea is, as with LCA described in 3.3.4, to allow finding concepts associated with a searched term through their cooccurrencies, or local contexts. The main distinction of LSA is the usage of *single value decomposition* for the *occurrence matrix*. Consider a text to be segmented as a set of sentences $\Delta = \{\delta_1, \delta_2, ...\delta_m\}$ with vocabulary $\{w_1, w_2, ...w_n\}$. Then the occurrence matrix of size $n \times m$ A is computed, where A_{ij} is the number of times w_i appears in δ_j normalized to the inverse document frequency. Then the single value decomposition of A is $A = U\Sigma V^T$; U, V are orthogonal matrices, Σ is a diagonal matrix and V^T is the transpose of V. Analysis of this decomposition yields the fact that the matrix AA^T is the word similarity matrix and the first k columns of U approximate it in a k-dimensional space, Λ_k (each column being a feature vector in the LSA space). In this way LSA "extracts the most important orthogonal dimensions, and, consequently, discards the small sources of variability in term usage. After this step, every word is represented by a vector of weights indicating its strength of association with each of the dimensions." [Bes06].

The benefit of this dimension reduction is not only lower complexity of the similarity function but also noise removal through omitting dimensions beyond k. This is possible due to the fact that LSA describes similarity of words through their context, or allocation, which is inherent to the very notion of topicality. LSA induces synonymic relationships between words but is even more useful since the same words and their synonyms can be used in different contexts, and LSA catches up for that by finding *semantic proximities* of terms. Further computation is conducted along the baselines of C99 [Cho00], using ranking and divisive clustering. The resulting error rate is reported to be down to P_k=8% on average and P_k=5% at best for longer sentences (with no stemming in the preprocessing step, Λ_{500}) on the artificial corpus. [Bes06] shows even better performance P_k=6.9% (compared to 9.7%) through training LSA on the whole and 25 times larger corpus.

A well-known improvement of LSA is PLSA which uses a probabilistic latent topic model [Hof99] similarly to LDA. Interestingly, an equivalence of PLSA and LDA (3.3.4) was shown for uniform Dirichlet distributions [GK03]. This was applied to topic segmentation in [BCT02]. A word can belong to more than one topic and this is expressed through a probability distribution of a latent variable connected to words of a document. Topic boundaries are estimated through a similarity measure of word probability distributions of two adjacent blocks; [BCT02] evaluates 5 different metrics. Topic boundaries are determined through thresholding. The method performed well on a news corpus with P_k=8.22% on average.

Finally, [ML07] presents a method coined GLSA (generalized latent semantic analysis) and argues that computation of the similarity matrix should involve the linguistic perspective. However, usage of *pointwise mutual information* (PMI) of two words does not outperform already described methods, resulting in P_k=17% on parts of the TDT corpus.

Decision trees

Decision trees are predictive models that can be used to classify data based on a set of describing features (Fig.3.16). An example of generally available algorithms used to generate such a tree from training data is *C4.5*.

Publications by [LP95] reported recall of 43% and precision of 63% and [MH06] P_k=28%. Significant better performance was reported by [DF99] (P_k=16.3%) and [PI10] (R=82%, P=80%). [DF99] obtained a very low false alarm probability down to $P_{falsealarm}$=0.09% by a reiteration removing boundaries of similar adjacent topics. [PI10] combined a machine learning approach with lexical chains as features.

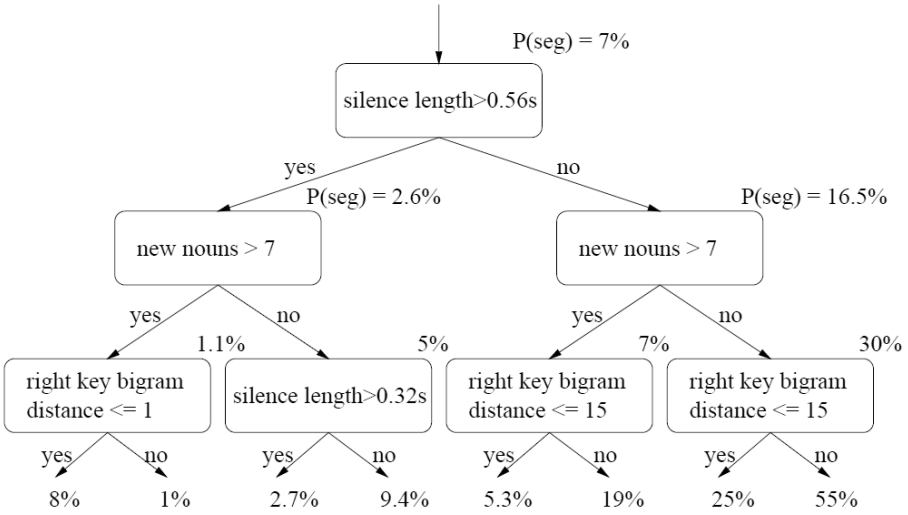

Figure 3.16.: *Example of a decision tree used to determine topic boundaries (top levels) [Dha00]*

.

Other approaches

[Rie02] proposed a probabilistic approach combined with a neural network obtaining a P_k-like metric value of 37.6% for an ASR meeting corpus.

Finally, a *genetic algorithm* was proposed by [Wu09] modeling topic segmentation as an evolution problem. The publication reported P_k down to 25.3% on the TDT-3 corpus.

3.4. Discussion

As already shown in Fig. 3.3 on page 22, there have been much research on topic segmentation since the early 1990s. A total of more than 40 publications indentified and presented in the course of this work explore a wide concept range. Since the early 2000s the research focus has shifted from relatively simple approaches like sliding window or dot-plotting towards more sophisticated machine-learning concepts employing mature theoretical foundations like SVM or LSA. This can be explained through growing understanding of managing speech data and information retrieval as well as decreasing hardware ressources costs allowing significant upscaling of experiments.

However, using a more complex approach does not guarantee better results if the algorithms does not fit the problem. As shown in Fig. 3.17 on page 44, the best results were achieved not only by machine learning approaches.

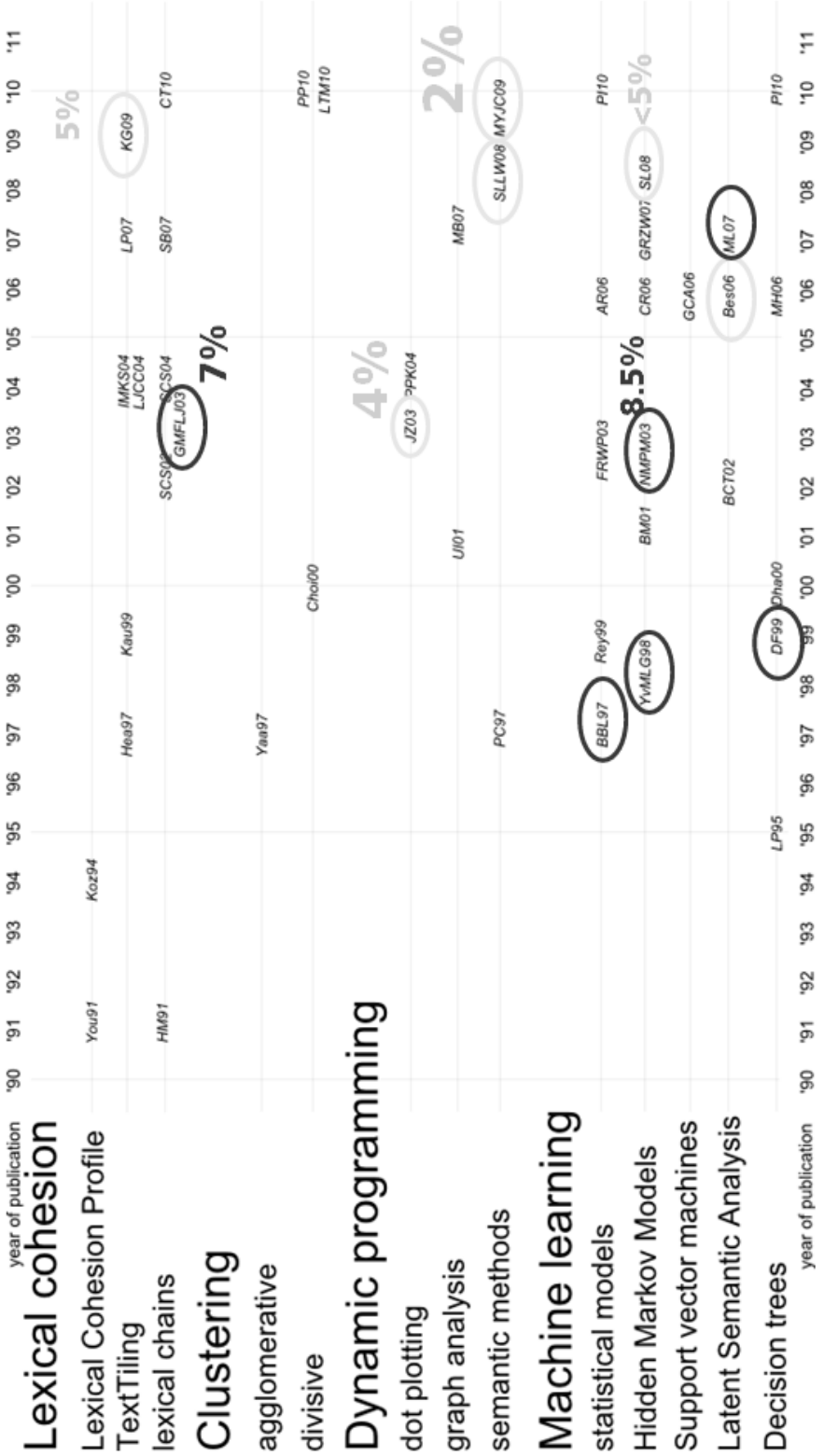

Figure 3.17.: *A chronological overview of publications reporting the best results. Legend: publications with $P_k < 25\%$ are labeled with ellipses (black: ASR; gray: text). $P_k < 10\%$ has explicit labeling with numbers.*

P_k	Author(s)	Summary
2%	Misra et al. [MYJC09]	DP, LDA, graph analysis *
4%	Ji et al.[JZ03]	DP, dotplotting, anisotropic diffusion
<5%	Fragkou & et al.[PPK04]	DP, dotplotting *
<5%	Sun & et al.[SLLW08]	LDA, Fisher kernel
5%	Kern & Granitzer[KG09]	TextTiling derivate
5%-7%	Choi et al., Bestgen [CWhM01; Bes06]	LSA dotplotting, ranking *
8%	Brants et al. [BCT02]	PLSA *
10%	Utiyama [UI01]	DP/stat. model;$P_k(ASR)$=35.2%[MB07]
10%	Peréz et al. [PP10]	overlapping incr. clusters

Table 3.2.: *An overview of the best topic segmentation methods with $P_k \leq 10\%$ applied to texts. Methods marked with asterisk * need training. DP stands for dynamic programming.*

The following two tables give an overview on methods which reported best results. A more detailed overview can be found in the Table B.1 (Appendix B).

So, Table 3.2 shows the best available topic segmentation methods having an error rate $P_k \leq 10\%$ (rounded to integer percent values) tested with normal texts. The comparison of previously presented methods relates only to their P_k values. Methods measured with the standard precision and recall metric are not considered due to the fact that these are mostly very early publications. Beyond that, P_k expresses the segmentation quality in a more adequate way (see 3.1.2) and is the mostly common used metric to compare topic segmentation performance.

Table 3.3 shows the best available results of segmentation algorithms applied to ASR transcripts (rounded to integer percent values, $P_k < 25\%$).

P_k	Author(s)	Summary
7%/32%	Galley et al. [GMFLJ03]	LCseg / lex. cohesion, lex. chains (TDT/ICSI)
9%	Matusov et al. [NMPM03]	section-level HMM *
12%-18%	Beeferman et al. [BBL97]	exponential models *
16%	Dharanipragada et al. [DF99]	decision tree *
17%	Matveeva, Levow [ML07]	GLSA
18%	Yamron et al. [YvMLG98], eval. by [NMPM03]	HMM *
21%-55%	Choi [Cho00] eval. by [GCA06; MB07]	dotplotting with ranking
22%	Georgescul et al.[GCA06]	SVM *
24%	Mohri, Weinstein [MMW10]	SVM *

Table 3.3.: *An overview of topic segmentation methods tested on ASR transcripts with $P_k < 25\%$. Methods marked with asterisk * need training.*

Both types of input, ASR transcripts or text, are preprocessed into stemmed word chains without punctuation or other marks by each presented approach before the segmentation step. In this way a written text looses its sentence and paragraph structure. Due to this fact, the only difference between these input types is the word error rate (WER) of the underlying ASR system. In terms of topicality the WER is signal noise which can reduce the topic relevance of a phrase through removing relevant words or which is even more worse introducing terms which are specific for some other topic. If an algorithm is sensitive to this issue, it can be expected to be erraneous to the extent induced by how high the WER is.

Although non-ASR tests can be intuitively expected to deliver better results due to absense of the ASR noise, there is not much experimental information on testing an algorithm on *both* type of data, using the same corpus. Only the few following publications shed light on this issue where the same algorithm was tested on the same data in both variants.

Despite the high word error rate (WER) of 24.5% [NMPM03] reports an increase of P_k of only 2% in case of ASR transcripts compared to manual transcripts (medical reports corpus). This observation renders this HMM-based approach to be robust to the WER, at least applied to medical reports. Also [CKGR05] states that WER up to 32% has little effect on topic segmentation quality in context of the news domain. However, both approaches exploit corpus-specific cue phrase topic indicators.

On the contrary to [NMPM03], a good example of ASR sensitivity is [UI01]. Tested with a set of 33 physics lectures, it performed with $P_k = 10\%$ on the manually transcribed corpus version, but on the ASR version its performance significally dropped to $P_k = 35\%$.

Thus, it can only be assumed that every algorithm was developed on specific data sets and its robustness could be limited even to interchanging the corpus due to unclear definition of what a topic is. For example, [GMFLJ03] performs with $P_k = 7\%$ on a news corpus, whereas testing it on a hard meetings corpus results in $P_k = 32-35\%$ [GCA06; SL08]. There is no clear evidence that topic segmentation evidence cannot be successfully transferred from one application domain (e.g. texts or manual transcripts) to another (e.g. ASR output).

To conclude, the above Tables 3.2 and 3.3 confirm the statement made by [LP08] (3.2): methods, performing well on artificial (Choi's corpus) or similar (TDT) corpora with sharp topic shifts, reflect this model to an extent that makes them not suitable for hard data like meetings. However, most adaptive methods could be in this case methods based on lexical component analysis (LDA, LSA), because they build on topic-specific word proximities rather than on similarity of adjacent regions.

3.5. Related work

This section gives a short overview on methods dealing with topic segmentation but concerning a multi-modal feature analysis which is beyond of the scope of this thesis.

Due to the fact that multimedia assets coming from the TV broadcasting asset expose their content on three different channels (visual, aucoustic and lexical), one can derive complex methods combining features from them all. The evolution of evolving methods has a natural connection to the development of available computing power.

One of the most earliest publications considering the multi-modal approach was proposed by [GS86], a theoretical framework describing annotation of speech with accentuation and phrasing. A pure acoustic analysis was done by [KIO96; SSHTT00] and [MPBG07].

[CHCC04; CKGR05] proposed hybrid approaches combining analysis of ASR output with either scene cuts or prosodic features.

[MHG+10] introduces combining of visual features with closed caption[3] analysis based on [MYJC09]. Finally, the most advanced methods combining ASR with both audio and visual domain features are to find in [May98; DR07; Pou09; GPHJ09].

[3] Closed captioning are additional subtitles originally concepted but not limited to the deaf community on the American continent. In Europe this is known as teletext.

4. Applied methods

This chapter deals with two unsupervised algorithms selected for the evaluation in the course of this work. Firstly, the corpora used for the evaluation are described, followed by a short discussion which known algorithms are most suitable for the targeted application. Finally, the baseline implementations and their performance are presented, followed by further analysis and experiments.

4.1. Used corpora

Data sets

Due to the limited public availability of exactly the same corpora referred in the previously presented research and the goals of this work being practical, efforts were made to create adequate application-oriented corpora for both test and development.

The test corpus consists of 13 *Tagesschau* German news broadcast transcripts collected online[1] about the turn of the year 2010/2011. The assets contain about 2100 recognized words and 11 topic segments on average. All assets have approximately the same length of 20 minutes except one short (about 4 minutes, which allows direct algortihm behavior evaluation during development time regarding short topic ranges). The reference segmentation pointed out to be parsable[2] from the broadcast navigation web interface provided by the broadcast station.

The development corpus, *DiSCo* (Difficult Speech Corpus) is provided by [BSB+10] and was constructed to be representative for challenges in German broadcast materials containing not only recordings of professional speakers but also spontaneous and dialect speech. The development corpus consists of 31 broadcast recordings with variable duration from 20 to 120 minutes. The assets contain about 5300 recognized words and 20 topic segments on average. The reference segmentation was elaborated through manual segmentation done with the help of an online annotation tool specially developed for this purpose.

[1] http://www.tagesschau.de

[2] GreaseMonkey is an add-on for the web browser Firefox. It allows runtime DOM processing. http://www.greasespot.net

Both corpora were transcribed by means of the *AudioMining* speech recognizer with estimated word error rate of 26.4% for planned speech to 51.2% for dialect speech as of [BSB+10]. The best WER=16.1% was measured for news broadcasts [BSB+09].

Human perception of topicality

The experiment in the course of the effort to annotate the DiSCo corpus with topic segments envolved about 50 human participants. However, not each segmentation was completed, so there are not enough cases to allow a representative study dealing with deviating annotations for topic boundaries of same assets. Still, the two examples given in Fig. 4.1 confirm the observations done in [Bal04]. In this case, there are only few boundaries common for the annotations done by at least the half of participants ($ZDF\#05$: four common boundaries, $ZDF\#04$: one common boundary).

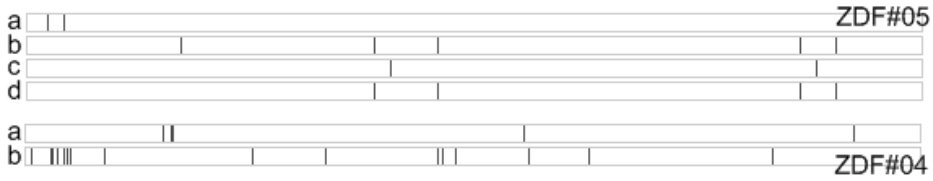

Figure 4.1.: *An example of how differently topic segmentation can be understood by different individuums. Two assets, ZDF#05 and ZDF#04 were annotated by 4 and 2 humans respectively.*

Analysis of these common boundaries reveals the four boundaries in the $ZDF\#05$ to be very clear scene cuts, identifying beginning of a new topic. Interestingly, the topic introduction by the main speaker was left out in both cases. The one common boundary in the $ZDF\#04$ is not really the same timestamp, there is a difference of 7 seconds and in both cases there is no objective justification for placing a boundary somewhere in the topic.

4.2. Method selection

This section discusses which methods from previously presented publications should be implemented and why. This thesis is focused on implementing unsupervised approaches.

Supervised approaches are in general more elaborative to implement since they need training. Besides this, most supervised approaches require a complex theoretical foundation which cannot be implemented in the course of this work. Finally, the investigation of recent research on topic segmentation has shown that both of these

approach types can yield good results: four unsupervised and five supervised approaches demonstrated $P_k < 10\%$ (Tables 3.2, 3.3). However, specifically for ASR results the lowest reported $P_k = 2\%$ on the artificial Choi's corpus was achieved by a supervised LDA approach [MYJC09] which can be considered for future work.

The remaining four unsupervised methods with $P_k < 10\%$ performed best in the news domain. Only one of them, the most influential lexical chains method LCSeg [GMFLJ03], was tested with ASR output and performed with $P_k = 7\%$. Lexical chains reflect the intuitive notion of topicality, especially in the news domain, because it seems quite natural to expect that distinct topics introduce coherent sets of keyword chains. This consideration renders LCSeg to be not the only one unsupervised ASR method with the state of the art word error rate, but also a perfect candidate for implementation in course of this thesis. However, it is to expect that as shown in [GCA06; SL08] this algorithm can show poor perfomance with $P_k = 32 - 35\%$ on a hard corpus like ICSI (meetings transcripts). However, if trying to segment a single meeting or many concatenated meetings on similar subject, a lexical chains based approach can expectably fail because of absense of clear separable dense chain sets in this case.

The remaining three non-ASR supervised methods [JZ03; SLLW08; KG09] with the state of the art word error rate have shown similar performance $P_k 4 - 5\%$. However, [SLLW08] was tested only with Chinese language and there is no clear evidence about language interchangeability for this algorithm. [JZ03] was tested with more data which is to some more extent artificial than [KG09] because Choi's corpus uses only document fragments. This renders [KG09] to be another good candidate for this work, also taking in account that its implementation and test is much easier than of [JZ03] which implies anisotroping filtering. However, [JZ03] is the best dot-plotting approach representative and should be also considered in future work.

The two next-best unsupervised methods are [UI01] and [PP10], both with $P_k = 10\%$. However, these approaches were developed and tested not in the broadcast domain. [UI01] was evaluated with a set of 33 physics lectures in their manual transcript version, showing P_k of 35% with the ASR version. This probably demonstrates a high ASR sensitivity of this method but due to the hard corpus it still could be promising for the broadcast domain. Finally, [PP10] deals with scientific publications. This type of data is very specific and therefore not applicable in this work.

In this section two candidate unsupervised methods for implementation in the course of this work were determined, TSF [KG09], derived from the TextTiling [Hea97] method, and LCSeg [GMFLJ03]. Both of them base on the language phenomenon of lexical cohesion.

4.3. TSF

This section deals with the unsupervised topic segmentation method TSF proposed by [KG09].

4.3.1. Baseline implementation

TSF [KG09] is a sliding window approach, detecting similarity fluctuations in a series of adjacent sentence blocks created from the investigated transcript. It builds to a large extent on the original sliding window method called TextTiling by [Hea97] (3.3.2). As shown in Figure 4.2, a transcript (a) is stemmed (d) and used to move a sliding window over it (d,e). The sliding window consists of two blocks and each position of it (*pos_i*) defines the measuring point for the similarity metric. The block size in sentences and sentence size in words (Fig. 4.2b) are user parameters, which should reflect the minimal length of desirable resulting topic segments.

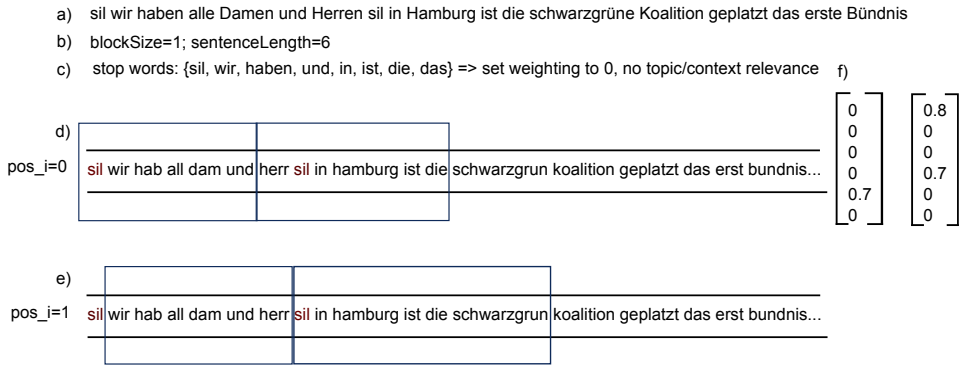

Figure 4.2.: *Details on the TSF algorithm: a) transcript fragment; b) parameters; c) fragment stopwords; d,e) sliding window; f) sentence TF-IDF vectors.*

In this case actually *dissimilarity* between the left and the right blocks is determined; on contrary to [Hea97], not graph valleys but peaks should be considered as boundary candidates.

The dissimilarity measure is defined as $\frac{sim_i^{inner} - sim_i^{outer}}{sim_i^{inner}}$ for each sliding window position i. Inner similarity is the average of all pairwise sentence similarities in left and right block on their own. Finally, outer similarity is again the average of pairwise sentence similarities for pairs taken from the left and the right blocks.

The similarity of two sentence corresponds to the cosine measure, or TF-IDF of two sentence vectors (Fig.4.2f). A sentence vector is a sequence of TF-IDF values for each word in the sentence, and the cosine measure can be interpreted as dot producrt of this vectors. More similar vectors result in a very small dot product. The TF-IDF values, or weights, are calculated based on word frequencies (TF) in the current

transcript and their IDF values resulting from the corpus. Weights of stop words are set to zero (Fig.4.2c) due to the assumption that common words like prepositions do not carry significant topical information being uniformely distributed over the transcript.

Topic boundaries are than hypothesized by taking a simple threshold which is a user parameter. Regions with many adjacent candidates are reduced by taking the candidate with the highest measured dissimilarity from its close neighbourhood.

The only significant difference of the implementation of this algorithm made in the course of this work from the original TSF is the selection of stop words. A common list of stop words was used, while TSF applies a complex routine to eliminate stop words by thresholding their dispersion in the transcript. However, for stop words have very little IDF weights it can be assumed that this should not have a great impact on the segmentation quality.

In [KG09] TSF was tested on the large RCV1 corpus containing about 800.000 news documents, resulting in $P_k < 5\%$.

4.3.2. Baseline results

The baseline TSF algorithm was tested with threshold 0.7 and block size of 4 sentences as proposed in [KG09]. Sentence length was assumed to be 16 words. With this settings, the test run yielded results shown in the Table 4.1.

Corpus	$P_k, \%$	$WD, \%$	**Ref. avg. count**	**Seg. avg. count**
news	54.9	61.5	11	4.8
DiSCo	50	57.2	20	5.6

Table 4.1.: *Baseline TSF results [KG09].*

Figure 4.3 renders the workwise of the TSF algorithm, showing the reference segmentation (a, bold continuous lines), TF-IDF word weights (b), and the dissimilarity function graph (c).

Figure 4.4 renders the workwise of the TSF algorithm, showing the reference segmentation (a, bold continuous lines), the final segmentation (b, thin continuous lines), the dissimilarity function graph (c) and the threshold (d). From this last figure can be observed how the algorithm selects boundary candidates.

It is apparent that there is that there is no definite correlation between regions with the highest peaks and the reference boundaries. We remeber that P_k defines the k sliding window parameter to be as big as the average of reference segment length.

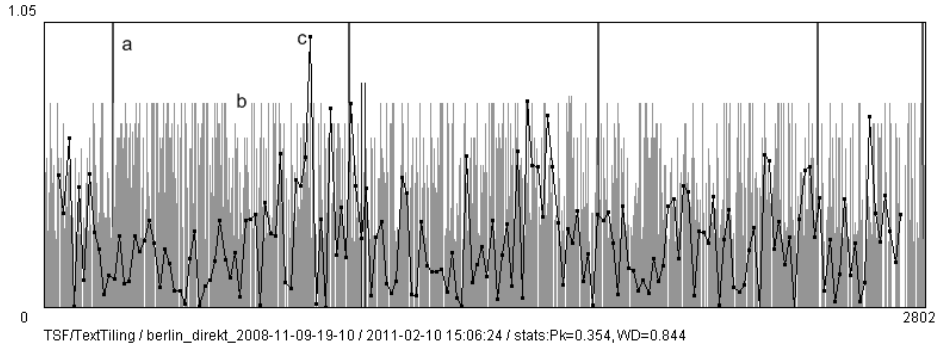

Figure 4.3.: *An example of TSF execution on an asset from the DiSCo corpus: a) reference segment boundaries; b) TF-IDF word weights; c) dissimilarity graph. Axis x: transcript words; axis y: normalized dissimilarity.*

Due to this fact, particularly this asset is qualified with a P_k value of 35.4% which is better than the corpus average of 50% in the course of this experiment, because 7 of 8 proposed topic boundaries are very close to the reference boundaries.

Using the adapted precision and recall metric proposed in 3.1.1 with the tolerance radius of 50 words, we can count TP=2, FP=4 and FN=3 (resulting in low values for precision and recall $P^* = 33\%$ and $R^* = 40\%$).

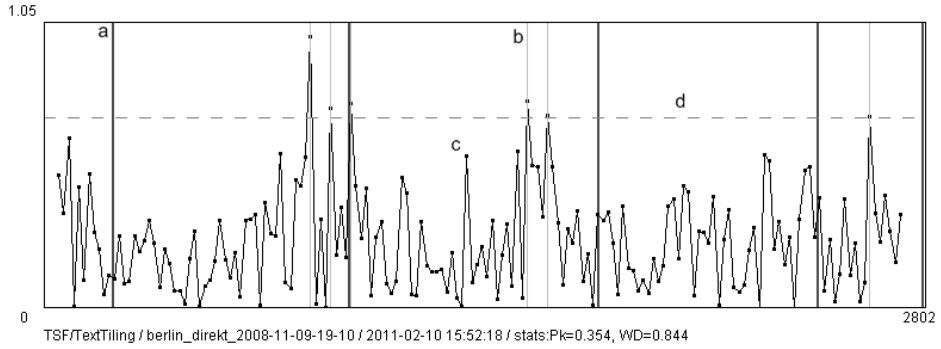

Figure 4.4.: *An example of TSF execution on an asset from the DiSCo corpus with final segmentation: a) reference segment boundaries; b) finally proposed topic boundaries; c) dissimilarity graph; d) threshold. Axis x: transcript words; axis y: normalized dissimilarity.*

4.3.3. Conclusion

Algorithms like TextTiling and TSF base on the assumption that due to lexical cohesion word distributions around topically coherent transcript segments would contain fluctuations around topic shifts which corresponds to vocabulary changes. However, handling words only by their frequency weights means dropping the semantic component of lexical cohesion.

In the course of this work, rather small-sized corpora were used compared to the original publication (13-30 vs. 800.000 assets). This leads to a limited vocabulary size and in effect such IDF weights distribution which probably does not allow sufficient emphasizing of topically coherent words. In other words, there are too many words with too similar weights (Fig. 4.2).

Some tuning was done on TSF like changing smoothing filter behaviour, but without significant performance improvements. However, a post-processing step similar with [DF99] to identify and eliminate false positives through counting common words shared by adjacent segment reduces the P_k down to 26%.

Due to poor baseline implementation performance and semantically less relevant algorithm concept TSF is not further investigated in favour of the next method, LCSeg.

4.4. LCSeg

This section deals with the unsupervised topic segmentation method LCSeg proposed by [GMFLJ03], based on the lexical chains model.

4.4.1. Baseline implementation

The baseline implementation following the original publication involves the same preprocessing steps as it is done in the most topic segmentation algorithms: from the tokenized document the stop words are sorted out and the remaining tokens are stemmed. Then, for each term used in the document-specific vocabulary a chain reflecting the term's usage throughout the document is created. This overall chain is saved as a series of smaller chains, resulting through breaking it on its weak points. Such weak points are characterized by gaps between single repetitions. The criterion, which determines a gap (also called *hiatus*) to be too large and hence enforcing a breaking point, is defined by a user-parametrized threshold.

[GMFLJ03] provides no information on how to handle terms which cannot build chains due to the fact that there are only single occurences of them. The baseline implementation assumes that these terms will not be tracked over topic segments (the algorithm does not consider semantic relations between terms) and are not considered.

The lexical cohesion function is computed over the chains as previously described in 3.3.2 on page 28. The idea is that terms usages should correlate with topic segments, because different topics normally introduce diverse subject vocabularies. If this is not

the case, this would mean that the topics are related in some way, and the (reference) boundary is probably not absolutely justified. In the lexical chains domain, this assumption means that multiple chains' starts and ends signalize topic boundaries.

A weight of a chain is determined by the number of terms it contains and its compactness. This weighting scheme gives higher scores to dense and short chains rather than larger to weaker chains. Following this logic, regions with high lexical cohesion should be consistent with dense and short chains allocations.

Finally, by reason that the local minima of the lexical cohesion function are just potential boundaries, further hypothesis is applied. A segmentation probability depending on the sharpness of the function values is computed which is eventually thresholded (only for probabilities higher than the another threshold, p_{limit}) with a simple, parametrized statistical metric $\mu - \alpha \cdot \sigma$ (α is a user parameter, μ and σ are the average and the standard deviation respectively).

4.4.2. Baseline results

The baseline LCSeg algorithm was tested with the maximum hiatus threshold of 11 sentences and block size of 2 sentences as proposed in [GMFLJ03]. Sentence length was assumed to be 16 words. Following the original publication, the parameters α and p_{limit} were set to 0.5 and 0.1 respectively.

Table 4.2 shows the results of the baseline LCSeg test run with this settings.

Corpus	$P_k, \%$	$WD, \%$	**Ref. avg. count**	**Seg. avg. count**	$P^*, \%$	$R^*, \%$
news	33.2	47.1	11	5.4	73.7	41
DiSCo	44.9	58	20	14.1	32.4	24
TDT ([GMFLJ03])	6.95	9.0	-	-	-	-
ICSI ([GMFLJ03])	31.9	40.4	-	-	-	-

Table 4.2.: *Baseline LCSeg results [GMFLJ03].*

Figures 4.5 and 4.6 render the workwise of the LCSeg algorithm applied to an asset from the news corpus. Thick lines display the reference annotation. In Fig. 4.5 the most sharp lexical cohesion graph valleys correpond to potential topic boundaries (thin lines).

Fig. 4.6 shows lexical chains found in the asset; in the top region there are longer chains reflecting terms with higher document frequency.

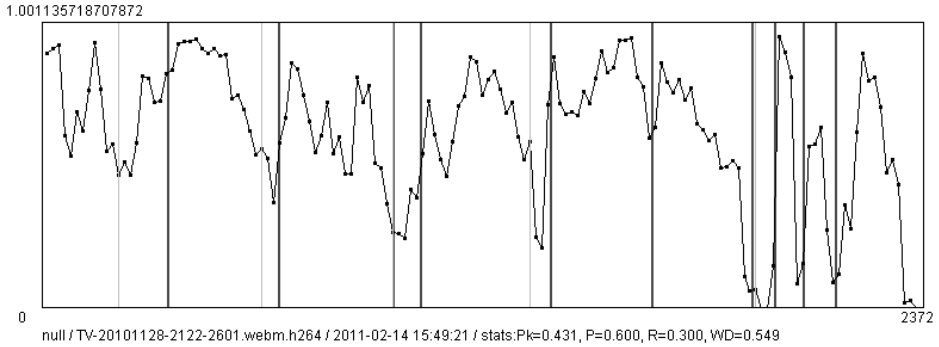

Figure 4.5.: *An example of LCSeg execution on an asset from the development corpus. Axis x: transcript words; axis y: normalized dissimilarity.*

4.4.3. Analysis

As shown in Table 4.2, the baseline implementation of LCSeg performs better than the baseline implementation of the TSF algorithm (Table 4.1). The error rate $P_k = 33 - 45\%$ is comparable to $P_k = 32\%$ measured with the hard corpus ICSI but worse than $P_k = 7\%$ measured measured the TDT corpus. Due to similar structure of the TDT and the news development corpus in this work it can be assumed that for the news domains the algorithm can gain better results after problem analysis and corresponding tuning. The differencies in performance in the news domains might relate to stronger cohesiveness of the TDT data segments. The WD error rate is however worse for both similar corpora, probably due to higher sensitivity of WD for different boundaries counts between reference and hypothesis segmentations.

The relatively high average precision rate $P^* = 73\%$ (within the range from 57% to 87% for the news corpus) along with the recall rate of $R^* = 41\%$ supports the assumption that the algorithm finds many close matches, producing not too much false positives (22 in the whole news corpus), but misses many reference boundaries. The low recall rate, $\frac{TP}{TP+FN} = 41\%$, shows that more than a half of positive boundaries is missed (the adapted precision and recall metrics use a tolerance window of 50 words to allow good scoring for close matches). Therefore, missed boundaries is the first problem that should be addressed.

As can be seen from the Figures 4.5 and 4.5, the first reference boundary is not found. This corresponds to a high value of the lexical cohesion function (LCF) at this point and simultaneously to multiple chains cutting the position of this reference boundary. This observation is however trivial because the LCF directly depends on overlapping chains by its definition.

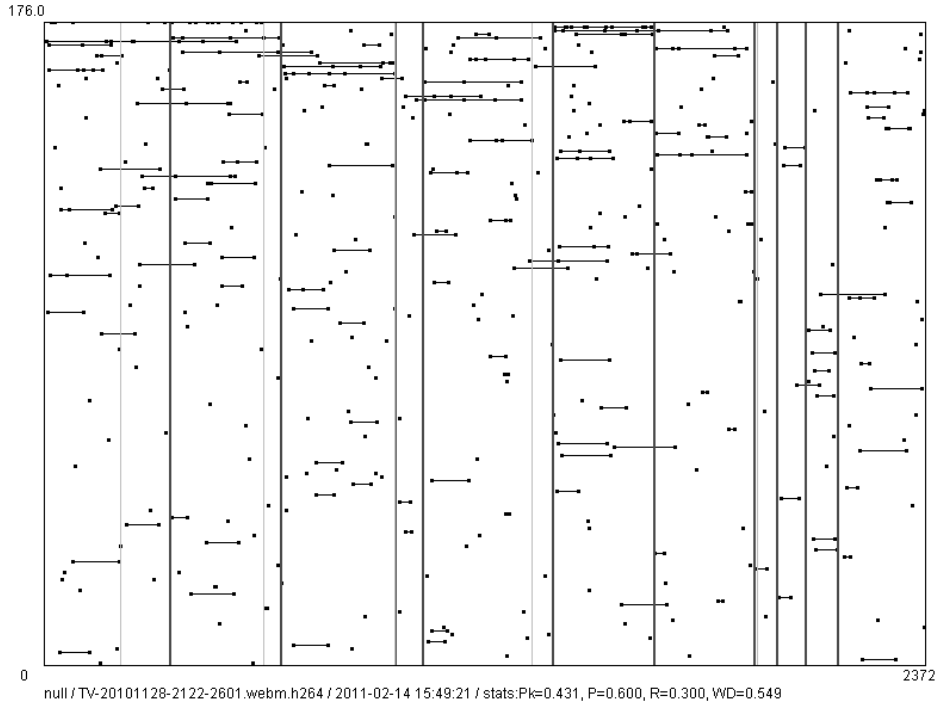

Figure 4.6.: *An example of LCSeg execution on an asset from the development corpus. Axis x: transcript words; axis y reflects found chains with growing term frequency frequency.*

In other words, chains connecting two adjacent segments are potential evidence of their topical coherence. More chains correspond to more cohesiveness, which reflects the fact that the same words are used in both segments.

However, topical coherence is strongly limited to terms used in coherent context. Chains connecting irrelevant terms result in misleading cohesion assumptions, which means missing reference boundaries.

Table 4.3 shows six chains cutting the first missed reference boundary of the asset shown in Fig. 4.5. To improve legibility, the transcript fragments are taken from the original transcript. The IDF values in the corpus range from 1.95 (*Grad*) to 4.75 (*Ausländerrecht*), where low values reflect frequent words and the highest values correspond to very rare words. The given example shows that at least in the news domain there can be less frequent words measured by IDF but they still can be scattered over adjacent topic segments even if there is no contextual connection. The most prominent example is the word *letzten* which is quite rare (IDF=3.17) even in its stemmed form, but still carries less context information.

If there are some generalization rules appliable to at least a part of words, which tend to build boundary-cutting chains leading to boundary misses, they could be used to find and eliminate these words to improve the performance of the algorithm.

No.	Left link	Right link	corpus IDF
1	ein *SPD* Bürgermeisterkandidat	*SPD*, FDP und Linkspartei	2.170
2	auch *wirklich* wollen	*wirklich* den gescheiterten	2.433
3	die *macht* im Bereich	haben wollt *macht*	2.585
4	*Hamburgs* Sozialdemokraten	Neuwahlen in *Hamburg*	2.948
5	hat die *Grüne* Basis	*Grüne* in Berlin	2.948
6	die *letzten* Umfragen	in den *letzten* Monaten	3.170

Table 4.3.: *LCSeg misses topic boundaries if the corresponding adjacent segments have enough words in common. The listed chains of a missed boundary are sorted by their terms' IDF.*

However, removing too much words would decrease the overall number of chains and the LCF scoring in general, leading to more false positives.

Table 4.4 shows the distribution of matches and misses in the news corpus regarding whether they are cut by chains or not (there are 145 reference boundaries in total in the corpus).

	cut	uncut
TP	53	8
FN	67	17

Table 4.4.: *LCSeg: distribution of cut and uncut boundaries over TP and FN. Most of false negatives are overlapped with chains.*

However, also most true positives were found despite of the fact that there are chain overlappings there as well. Table 4.5 shows a detailed comparison of all four reference boundaries subsets.

	count	total chains	avg. chains	avg. IDF
TP+cut	53	111	2.09	2.67
FN+cut	67	159	2.37	2.38
FN+uncut	17	-	-	-
TP+uncut	8	-	-	-

Table 4.5.: *LCSeg: distribution of the 145 corpus reference boundaries as cut and uncut matches or mismatches.*

The differences between misses and matches for cut reference boundaries are subtle but still obvious:

- a miss is cut by more chains at average than a match;
- IDF weight of chains cutting a miss is lower (more frequent words).

We can follow that finding boundaries must be sensitive to the balance of inner-segment and boundary-crossing chains. Segments having not enough dense inner chains but too much words in common with adjacent segments are merged with them, leading to misses. This explains large number of matches which are also cut, but with less chains and by less rare words.

Further investigation of the correspondces between the terms leading to the boundary-cutting chains (Fig. 4.7) shows that many of these terms, responsible for the most cuttings, really belong to more frequent words. However, there are also many single boundary-cutting chains of words which are rather rare.

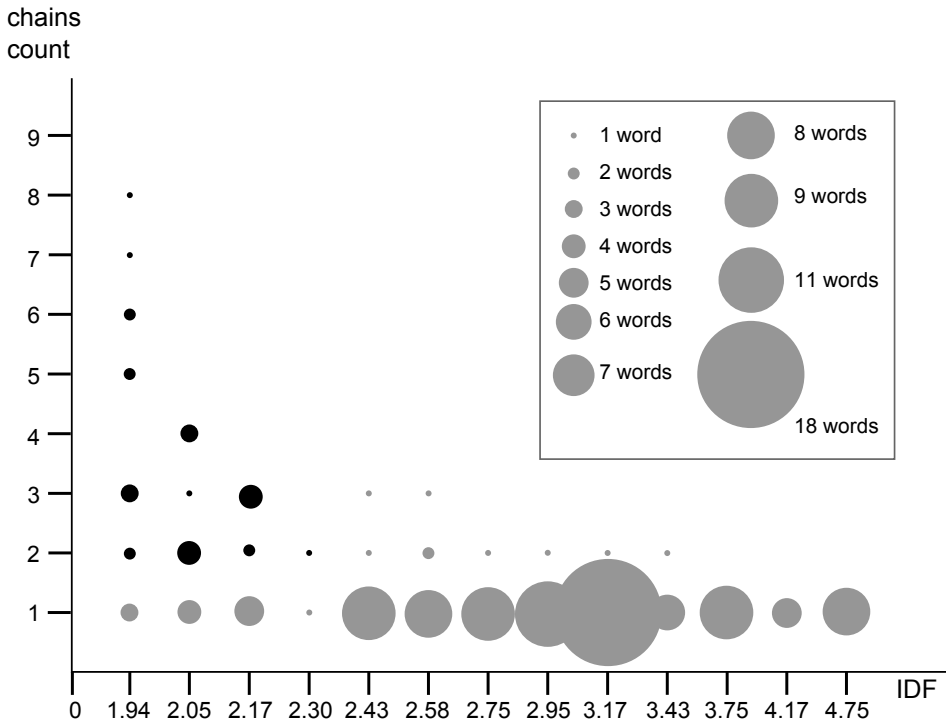

Figure 4.7.: *The distribution of boundary-cutting chains shows that most cases happen due to most frequent (black) and most rare words. A circle of radius n depicts different terms count as a function of a specific IDF (axis x) and corresponding number of boundary-cutting chain cases (axis y).*

The first tendency can be easily explained by the fact that the stop word list has a general character, containing context-free words like prepositions and articles. However, some other words like *wirklich* do not carry specific information and can be added to the stop word list. Also regional and politics-related terms (or their IDF representants) like *Deutschland, Hamburg* (IDF=1.9, 2.9) or *SPD*(2.170) might be common to many news topics in German news, being relatively unspecific. The second tendency reveals many rare words like *Grundlage* (3.76) or *vorbereiten* (4.76), however, there are only single cases where they occur as boundary-cutting chains.

The impact on misses in this case is that rare words have serious influence on the LCF through their high IDF weight.

In-depth analysis of all 67 false matches for cut reference boundaries in the news development corpus reveals that there are different problem classes shown in Table 4.6 leading to creating chain links where they should not be.

item	problem class	occurences
P1	accidental occurences in unrelated local contexts	71
P2	too long chains	45
P3	chaining of topically unrelated terms caused by stemming	17
P4	ambiguous boundary between topically related segments	16
P5	bogus chaining caused by ASR errors	8
P6	homographs	0

Table 4.6.: *LCSeg: analysis of boundary-cutting chains at boundary misses*

P1 Accidental occurences of terms in adjacent segment but in unrelated contexts are the most frequent problem. Consider the English word *party*. If there are two adjacent news segments about politics, referring to issues in different parties, these are actually two different topics, but the LCF will be higher at the boundary between them and contribute to a possible false negative. These accidental occurencies cannot be identified alone by their IDF values, because there are many terms with relatively high IDF but as said above too topically unspecific. Good examples of news domain terms being topically unspecific but rare are *Wirtschaft* (economy, IDF=3.17), *Schneeverwehungen* (snow banks, IDF=3.433) and *Spanisch*(related to Spain, IDF=3.755). Despite of the high specifity of these words, German news report on events in countries and economies, or there are indirectly related segments, merged through a very high IDF weight of just one common word. This problem class renders the weakness of LCSeg as an approach based on scoring simple terms repetions, which is unable to recognize different local contexts. LCSeg would not be always able to separate two topics reporting about youth, education and schools and about youth, schools and a regional amok accident, especially if both of them mention same region. Also the assumption that different topics have almost no words in common is not completely correct just due to the fact that news relate to repeating collections of terms, making accidental matches more common than might be expected.

P2 Too long chains, linking non-adjacent segments (Fig. 4.8) can be partly suppressed by reducing the hiatus parameter, so that chains cannot be linked over too long range. However, the problem is that some topic segments are far too short un-

der the average length, e.g. the average words count of a news reference segment is 190 words, and the shortest segments can contain only about of 20 words.

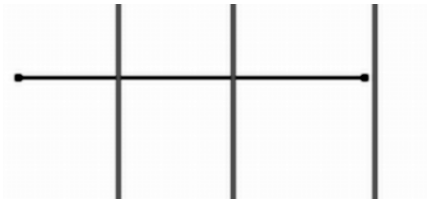

Figure 4.8.: *Too long chains can link neighbours of a short segment, suppressing the boundaries.*

P3 In this case one important drawback of stemming algorithms should be emphasized. It would be more adequate to take an algorithm, reducing a word to its uninflected form, because reducing words to their stems can lead to chains build between topically or even semantically unrelated words with different stems. Examples are *Jugend - Jugendlicher*, *Grund - Gründer* and *neu - neun*. Deactivating stemming would probably make many chains disappear because single word flexions cannot be expected to be always frequently repeated in one topic.

P4 A reference boundary can be seen as ambigous, i.e. not completely justified, if two segments handle the same event and their common words are used in the same context. This problem class is hard to address because it is related to our understanding of topicality. In such cases, there are actually two subsequent segments on the same topic or probably subtle subtopics of it. Still, a segmentation algorithm would fail to recognize an ambiguous boundary if it handles only transcript-intrinsic indicators.

P5 ASR errors can also cause incorrect chaining. However this problem class is not very frequent for probability reasons, at least with relatively low word error rate: it is not to expect that one more or less random ASR substitution error caused through background noise (e.g. a car) or unclear pronunciaion (e.g. mumbling) would match to a close word of an adjacent segment. However, the probability of random false matches can be addressed through constant training of the ASR system for recognizing up-to-date terms. Good examples are *WikiLeaks* transcribed as *Mitglied*, and *Präsident Medwedew* transcribed as *Präsident mit Erde*.

P6 An expected but not observed problem class are homographs. There can be two completely incoherent topic segments, e.g. using homographs of the word *party*. For example, *the bill got the most votes from the Republican* ***Party***. *And now the local news: a 40-year-old man was shot to death late Friday night at a birthday* ***party*** *in Lake County*. The algorithm (which works with the stemmed and downcased version of the transcript) would give the LCF a high score for the chain consisting of the both *party* occurences, completely ignoring the difference between birthday and

political parties. Apparently, the probability for homographs occuring in adjacent news segments is very low.

In the course of these observations one another fact was observed. There are many cases where a word occurs only two or three times but is not linked to a single chain segment because of big distances between single word occurencies. Despite this fact, these one-word chains also have their influence on the LCF. Therefore allowing chains of words occuring only once in the whole transcript might be helpful to emphasize inner cohesion of reference segments.

4.4.4. Experiments

The analysis of the behaviour of LCSeg tending to miss many reference boundaries allows assumption how this problem can be tackled by addressing specific problem classes. The Table 4.7 gives an overview of experiments done to investigate the test feedback on handling the observed problem classes.

item	**problem class(es)**	**action**	**expectation**
E0	information loss	allow single-word chains	higher cohesion inside reference segments
EA	low IDF, P1	suppress chains with low IDF	less unimportant chains
EB	high IDF, P5	suppress chains with high IDF	less outlier weighting
EC	unspecific terms, P1	extend stop words list	semantic suppression of context-free chains
E2	P2, P1	decrease hiatus	suppress links between non-adjacent segments
E3	P3,P1	disable stemming	avoid links between unrelated terms
EX	algorithm design	introduce logic to increase boundary probability at places with long non-speech	performance improvement through additional clues

Table 4.7.: *LCSeg - experiments proposals.*

The problem classes **P1** and **P5** can be addressed only in an indirect way, because **P1** reflects the weakness of LCSeg and **P5** depends on the ASR word error rate which is an external influence. So, accidental matches can be generally avoided through any approach leading to finding shorter chains at the right places. Occasional links for word pairs containing ASR errors can be probably reduced through removing

words with the highest IDF values. The problem class **P4** cannot be addressed due to algorithm design.

The experiment **EX** introduces an extension to the algorithm. It might be possible to diminisch boundary misses through using additional information like longer non-speech segments or speaker changes. Analysis of the news domain shows that speaker changes have high occurence which does not correlate with topic boundaries for the reason that most topics are introduced and closed by the same speaker. Long silences are common but not exclusive to introduction of topic changes. A specific problem of the *AudioMining* system is in this case that it creates *sil*-markers for both non-speech and silence segments.

exper.	P_k *nc*,%	WD *nc*,%	P^* *nc*,%	R^* *nc*,%	P_k *DC*,%	WD *DC*,%	P^* *DC*,%	R^* *DC*,%	arg
baseline	33.2	47.1	73.7	41.1	44.9	58	32.4	24	
E0	35.4	48.0	71.2	36.2	44.3	57.1	33.5	23.9	
EAa	37.6	49.7	69.5	37.2	42.5	57.2	35.4	25.2	idf>2.0
EAb	40.2	49.4	67.9	30.5	44.6	55.6	30.3	21.0	idf>2.3
EBa	38.1	50.6	71.2	32.8	46.1	57.3	31.6	20.7	idf<3.2
EBb	38.1	50.6	71.2	32.8	46.3	57.5	31.2	20.9	idf<3.4
EC	40.2	49.5	69.0	34.1	42.8	54.3	37.7	20.9	
E2a	47.5	54.9	49.6	19.0	46.7	54.5	30.1	13.8	h=80
E2b	41.9	52.2	55.0	29.6	44.1	56.2	31.7	21.4	h=120
E3	33.6	48.4	72.9	42.2	42.9	55.9	32.8	23.5	
EX	30.9	51.6	61.3	54.6	44.9	58.0	32.4	24.1	

Table 4.8.: *LCSeg - experiments results (nc stands for the news corpus and DC stands for the DiSCO corpus).*

Contrary to the expectations, none of the experiments following the original algorithm design yields a significant improvement. The baseline implementation could not be outperformed clearly, e.g. slightly better precision leads to a worse recall rate. However, the baseline results on the hard DiSCo corpus (P_k =44.9%) are not much worse than the ICSI corpus results ($P_k = 35\%$).

Table 4.9 summarizes observations of the details on the experiment execution.

4.4.5. Conclusion

It can be concluded that the LCSeg performance on the TDT corpus assumed correct implementation must rely on the artificial character of the TDT news segments. Real news data seems to exhibit no topic changes which are clear enough. The additional error source, the ASR word error rate, does not seem to play an essential role in this

experiment	**details**
E0	distribution of words with single occurences has random character
EA,EB	removing chains by IDF has no selective impact on the chain structure
EC	removing more terms weakens cohesive links inside segments
E2	changing of the hiatus parameter does not outweigh **P1**
E3	disabling stemming removes incorrect matches but has also a negative effect through breaking up adequate chains
EX	speaker changes are inadequate in news domain; long silences are mixed up with non-speech by *AudioMining*

Table 4.9.: *LCSeg - experiments details.*

context. It might be interesting to analyze real silent pauses, which are not mixed up with non-speech or unrecognized speech.

5. Implementation details

This chapter considers the requirements to the implementation of topic segmentation algorithms from the software architecture perspective. Firstly, the algorithms are analyzed, followed by the architectural concepts derived from the analysis results. The chapter concludes with a short review of the runtime behaviour.

5.1. Algorithm analysis

The first algorithm, TSF, uses the sliding window approach to iterate over the transcript. After some preprocessing and executing the core logic the algorithm calculates a set of values for the transcript dissimilarity function. These values are processed through a number of steps like thresholding, smoothing and selection to produce finally some sequence denoting topic boundaries.

A topic boundary is however domain invariant in the sence that it is valid for both transcript lexical and temporal domains. It is an offset position or a timecode in the timecode sequence, but only an offset position in the textual part of the transcript.

The second algorithm, LCSeg, performs in a similar manner, the sliding window and core logic responsible for the chains model output a sequence of the similarity function values. These values are also changed by smoothing filters and reduced to the topic boundary candidates set.

There is a need for both algorithms during development to measure their quality involving error metric algorithms and a reference segmentation. Both algorithms depend on a set of parameters which might need to be set depending on the data flavour.

Finally, a transcript can be acquired from different sources, for example from a file on the hard disk, which is very convinient for testing, or through the *Audio-Mining* webservice providing transcript data, which might be useful in a scalable production environment.

Topic segmentation algorithms will be wrapped to a webservice component and the execution results will be included into the XML transcript files. This implementation part has a very generic character and is not further described.

These facts can be summarized as the following:

- a transcript should be modeled as a source-independent instance;
- both topic segmentation algorithms use sliding window approach;
- both topic segmentation algorithms take a transcript as input and output similarity or disimilarity values as a function of the transcript;
- both topic segmentation algorithms use postprocessing steps in terms of signal processing, mapping the core logic output to a set of hypothesis topic boundaries;
- for extensibility reasons, it should be possible to interchange the algorithm as well as its configuration;
- a topic boundary can be defined in many ways, referring to the lexical or the temporal part of the transcript.

5.2. Architectural concepts

The software design concept will be presented along the lines of the MVC pattern [GHJV94], preceded by the middleware layer. The planned software module should be process-oriented and does not need an (interactive) view by given requirements. However, for the development phase it is often helpful to have some visual output, and structuring data and logic into controller and model components helps creating an extensible and transparent architecture.

5.2.1. Middleware

The middleware level in this case is a set of low-level service components, providing abstract access to different resources. Most important instances are transcript providers, the configuration processor and the global service object.

A transcript provider implements access to some concrete transcript ressource, for example a folder containing XML files or an *AudioMining* repository which can be accessed only through the webservice interface (Fig. 5.1).

The configuration processor allows a configuration-driven setup of the whole topic segmentation process. Being a bean factory inspired by the Spring framework[1], it instantiates and configures all objects instructed by an XML configuration file. A bean factory corresponds to the *abstract factory* design pattern [GHJV94], responsible for creating (Java) beans, which are simple objects with a default constructor and the interface limited to setters and getters. This implies all important objects

[1]Spring framework is a JEE framework providing best practices, http://www.springsource.org

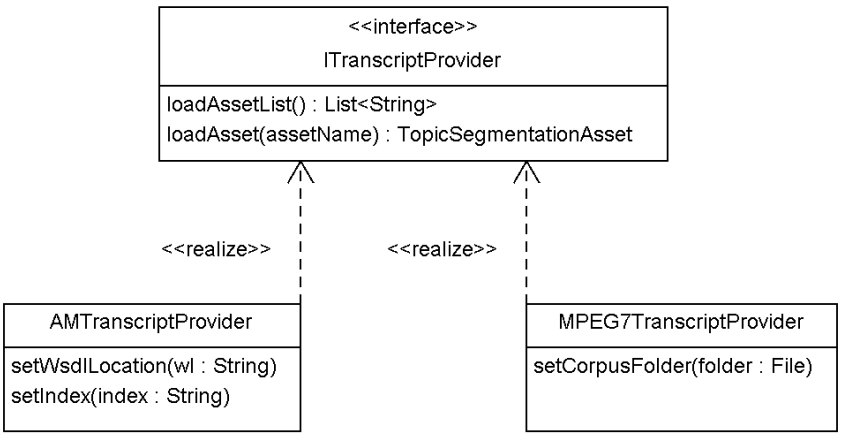

Figure 5.1.: *Interface and class diagrams for transcript providers.*

to be beans, which has appeared to be a good practice also from previous projects. The bean factory approach allows not only the parameters but also classes, including algorithms, to be easily interchangeable. The generic knowledge about object interface details in delegated to the XML configuration file. An example XML fragment defining an execution setup is given in the Appendix D.

Finally, the global service component following the singleton pattern [GHJV94] is thought to encapsulate other singleton objects like current IDF model or language stemmer which need to be instantiated only once. Singletons can tend to become a problem due to the need of serializing access to them in multi-threaded environments. For this reason it might be advisable to refactor these global services out to auxiliary webservices.

Worth mentioning is also the sliding window iterator. It enables access to the series of the sliding window positions over the iterator pattern [GHJV94], which allows for coding on a more abstract level without permanent indices references.

5.2.2. Models

Due to the fact that the transcript can be loaded from different sources and can optionally contain a reference segmentation, which also might be obtained from heterogeneous sources, a model of a *topic segmentation asset* (Fig. 5.2) appears to be adequate in this case. This object encapsulates all known facts referring to a transcript, like the timecode and word sequence as well as an optional reference segmentation. It provides different possibilities to access the facts (e.g. the segmentation as offsets array or an object sequence).

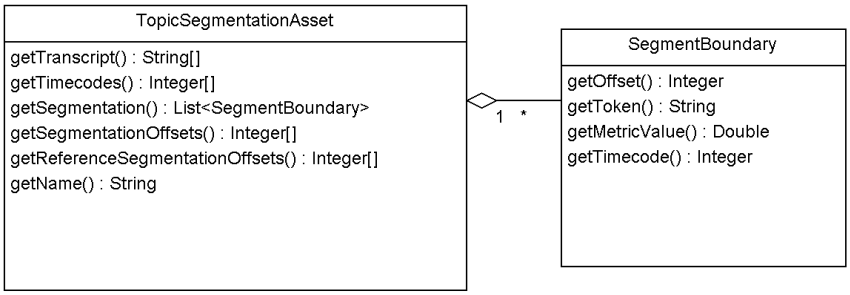

Figure 5.2.: *Class diagrams for topic segmentation asset and segment boundary. For legibility reasons only the most important getter methods are shown.*

A *segment boundary* (Fig. 5.2) which can be also a topic boundary is modelled as an abstract model to simplify access to the underlying data. A boundary is most useful as an offset position during processing and it can be easily mapped to the corresponding transcript locations in both lexical and temporal dimensions. It has also an associated metric value, which is a corresponding (dis)similarity function value measured at this place. This approach allows comfortable handling of different segmentations being a series of segment boundary instances, having constant access to all relevant data.

5.2.3. Controllers

There are four controller types in the topic segmentation module: the main controller and the core segmentation algorithm, followed by the post-processing filters and error metrics.

The main controller triggers the segmentation process initiation with the help of the configuration processor and executes this process on its highest abstraction level: the transcript provider delivers the segmentation assets and pass them over to the core segmentation algorithm, after which the assets undergo filtering through filters and evaluation through error metric algorithms (Fig. 5.3).

As already can be seen from this description, the main controller implements the topic segmentation process as a pipeline, following the *pipes and filters* software design pattern. This design pattern is generally known already from the Unix console scripts, where data can be passed over from one command to another. An execution pipeline does not provide optimal runtimes because of the multiple reiterations over virtually the same data in each component. At least during development time it is very helpful because it allows a quick rearrangement and reconfiguration of the software module.

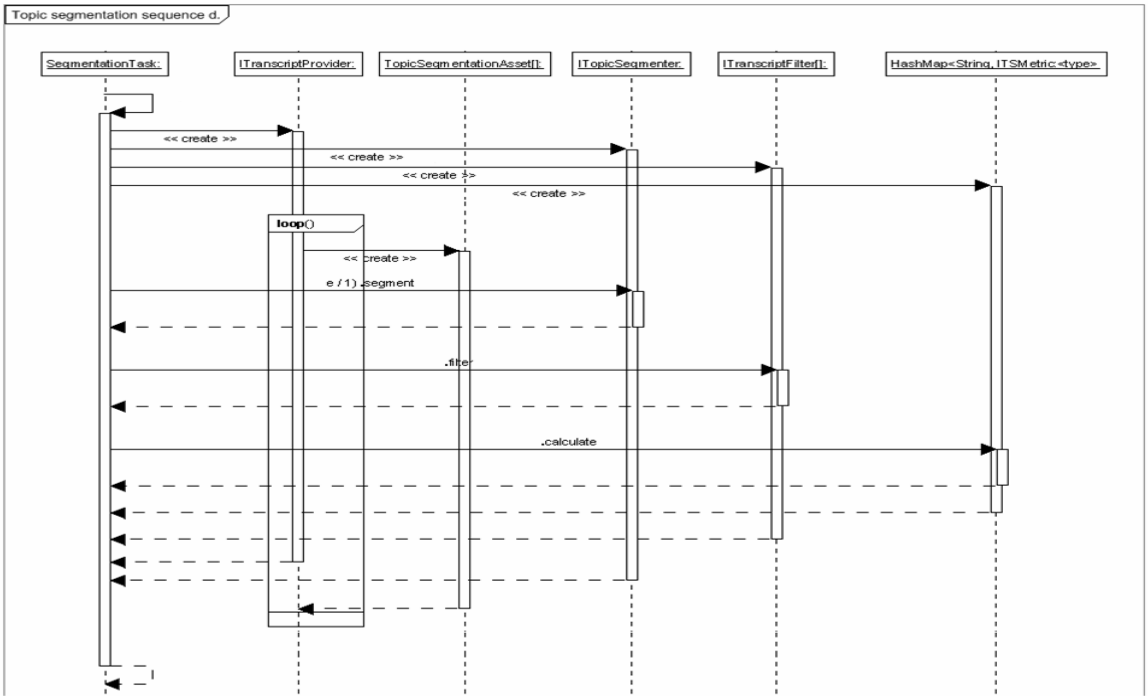

Figure 5.3.: *Sequence diagram of the topic segmentation pipeline.*

5.2.4. Views

As already said, views are not needed by the topic segmentation software by requirement because it is a software component needed in the backend environment. However, during development and test and also for research purposes views in the sense of creating execution reports were developed. These are charting classes for rendering diagrams 5.4 related to the segmentation process and a simple HTML generator to allow for exploring transcripts enriched e.g. with similarity function values or markers for boundary candidates along with the reference segmentation.

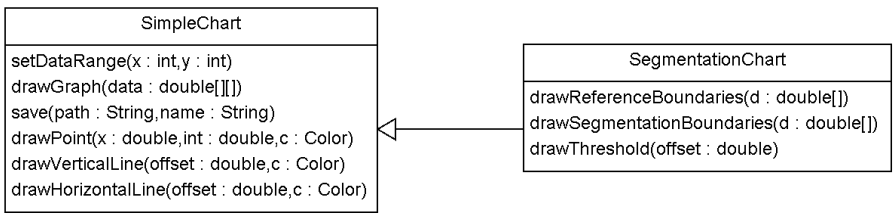

Figure 5.4.: *Class diagrams for the charting functionality.*

5.3. Performance

For the tests demonstrated in this work the following hardware configuration was used: a 4x2.8GHz CPU with 8 GByte RAM. Topic segmentation executed for the DiSCo corpus yielded processing rates of approximately 11230 words per second (an asset contains about 2-14K words). Due to the need to build the chain model LCSeg needs some more time; however, performance details has not been explored.

The runtime performance and hardware requirements can be optimized through better usage of disk and memory resources which was kept in mind but not primarily targeted during developement. Removing the pipeline excecution design after finding the optimal filters constellation might need some basic refactoring, yielding a linear runtime optimization.

6. Conclusion and future work

In this final chapter the previous findings are summarized. The chapter concludes with a short discussion and a prospect on the future work as well as an overview of the most comprehensive sources for further reading on topic segmentation.

6.1. Summary

One of the main challenges of the modern digital audiovisual media is the challenge to make vast data archives searchable. The spoken content of these media is transferred to the text domain by means of audio speech recognition systems. However, unlikely to written texts speech has no obvious logic structure. A key component in this area is topic segmentation, aimed to assist separating media by subject and enabling futher processing steps like text summarizing or subject-specific ASR reiteration.

The terms topic and topicality have a close relation to specific locality in the discourse, which is connected to the spatial view on texts. There is a broad variety of indicators for topic shifts, both intrinsic to the lexical domain and also transcript-extrinsic like transcript metadata (e.g. silences) or features extracted from the visual media component. However, none of these indicators are clear enough to determine a topic boundary. A further problem with the topic segmentation task is its ambiguity. This can be demonstrated through deviating segmentation proposals made by humans for the same assets.

Measuring quality of a topic segmentation algorithm cannot be done with the standard precision and recall metrics, because they do not take in account that close boundary matches are an evidence for good segmentation quality. For this reason special error metrics are used, in most cases this is the P_k error metric. The main backdrawal of the P_k metric is however that it does not penalize false alarms appearing in series.

Despite the fact that P_k has been widely used, a comparison of algorithms is a challenging task for the reason that there is no standard and publicly available test corpus which would allow solid benchmarking studies[1].

[1]For example, in the browser domain there is the Acid test (http://acid3.acidtests.org/), and in the 3D visualisation domain there are publicly available object models.

Since the early 1990s much research has been done on topic segmentation, however most of it does not relate specifically to ASR transcripts. Publications presented in this work explore a wide range of algorithms which can be used for topic segmentation. A very characteristic feature of research in this area is to use artificially compiled data sets for testing, which implicitely predefine clear topic shifts.

Two algrorithms were selected by a set of criteria and implemented. Both were tested with real data. The analysis of the results has shown that the ASR word error rate influencing transcript integrity presents a smaller problem than the loss of semantic and topical information introduced through algorithm design targeted at simplifying lexical cohesion down to single word usages.

6.2. Conclusion and future work

Investigation of the algorithms LCSeg and TSF has shown that there is a need for a model which would come closer to the topicality as such. Accidental matches of terms in adjacent topic segments were found to be much more frequent that it could be assumed. Especially the news domain seems to operate with recurrent term collections over different topics.

This means that analyzing single words is a loss of information in terms of topicality. Approaches extracting local contexts from the topics can be expected to perform better, because they emulate semantic relations between terms ([MYJC09]). Also the method proposed in [JZ03] is of interest because through the anisotropic diffusion topic segments are consolidated and the unsharp edges are removed, which corresponds to filtering out blurred topic transitions.

An important aspect related to topic segmentation which yet seems to be less investigated is sentence boundary extraction ([RR97; KL10]). Specifically with ASR transcripts it would be helpful to hypothethize topic boundaries on a set of previously proposed sentence boundaries defining a first-level logical structure in the transcript.

Another promising research direction is the multi-modal analysis. Especially long silence breaks should be annotated in the rich ASR transcript and taken into account.

6.3. Further reading

Due to the fact that the bibliography of this thesis contains many references, this section gives an overview on the most comprehensive sources on topic segmentation.

The most recent books providing good reviews on the topic segmentation problem and research are [Pur11](2011) and [GZ09](2009). Another book completely dedicated to topic detection and tracking [ALJ00] appeared in 2000. It is however a collection of technical reports from the DARPA TDT challenge, but it still gives a good idea of the early research and main ideas in this area.

There have also been three PhDs dealing with topic segmentation providing a good exploration on the subject by Reynar, Choy and Weinstein [Rey98; Cho02; Wei09]. Worth mentioning is also the BSc thesis [Bal04] by Ballantine.

Bibliography

ALJ00 James Allan, Victor Lavrenko, and Hubert Jin. Comparing effectiveness in TDT and IR. Technical report, University of Massachusetts, Massachusetts, 2000. 6, 8, 15, 73

Alp10 Ethem Alpaydin. *Introduction to machine learning*. MIT Press, Cambridge, 2nd revised edition, 2010. 36

AR06 Jaime Arguello and Carolyn Rosé. Museli: A multi-source evidence integration approach to topic segmentation of spontaneous dialogue. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 9–12, New York, 2006. 39

Ari89 Aristotle. *Topica*. Harvard University Press, Cambridge, 1989. 4

Aue96 Peter Auer. *Sprachliche Interaktion*. Max Niemeyer Verlag, Tübingen, 1996. 5, 7

Bü10 Fabian Bürger. Figure to the Wikipedia article Support Vector Machine. http://de.wikipedia.org/w/index.php?title=Datei:Svm_intro.svg, 2010. [Online; accessed 02.12.2010]. 41

Bal04 James Ballantine. Topic segmentation in spoken dialogue. Master's thesis, Macquarie University, Sydney, 2004. 8, 15, 49, 73

BBL97 Doug Beeferman, Adam Berger, and John Lafferty. Text segmentation using exponential models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997. 38, 45

BBL99 Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34, 1999. 16, 17, 18

BCT02 Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the Eleventh international conference on Information and knowledge management*, 2002. 42, 45

Bes06 Yves Bestgen. Improving text segmentation using latent semantic analysis: a reanalysis of Choi, Wiemer-Hastings, and Moore (2001). *Computational Linguistics*, 2006. 41, 42, 45

BM01 David M. Blei and Pedro J. Moreno. Topic segmentation with an aspect hidden markov model. In *Annual ACM Conference on Research and Development in Information Retrieval*, 2001. 39, 40

BSB+09 Doris Baum, Daniel Schneider, Rolf Bardeli, Jochen Schwenninger, Barbara Samlowskiy, Thomas Winkler, and Joachim Köhler. DiSCo — A speaker and speech recognition evaluation corpus for challenging problems in the broadcast domain. In *GSCL Symposium*, Duisburg, 2009. 49

BSB+10 Doris Baum, Daniel Schneider, Rolf Bardeli, Jochen Schwenninger, Barbara Samlowskiy, Thomas Winkler, and Joachim Köhler. DiSCo — a German evaluation corpus for challenging problems in the broadcast domain. In *Proceedings of LREC 2010*, Malta, 2010. 48, 49

CHCC04 Tat-Seng Chua, Winston Hsu, Lekha Chaisorn, and Shih-Fu Chang. Story boundary detection in large broadcast news video archives: techniques, experience and trends. In *12th annual ACM international conference on Multimedia*, 2004. 47

Cho00 Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. *ACM International Conference Proceeding Series*, 4:26 – 33, 2000. 30, 31, 33, 34, 35, 42, 45

Cho02 Freddy Y. Y. Choi. *Content-based Text Navigation*. PhD thesis, Department of Computer Science, University of Manchester, Manchester, 2002. 7, 8, 13, 73

CKGR05 Heidi Christensen, Balakrishna Kolluru, Yoshihiko Gotoh, and Steve Renals. Maximum entropy segmentation of broadcast news. In *Proceedings of ICASSP 2005*, 2005. 46, 47

Cor02 Leiserson Charles E. Rivest Ronald L. Cormen, Thomas H. *Introduction to algorithms*. MIT Press, Cambridge, second edition edition, 2002. 7

CR06 John F. Canny and Tye Lawrence Rattenbury. A dynamic topic model for document segmentation. Technical Report UCB/EECS-2006-161, University of California at Berkeley, Berkeley, 2006. 40

CST10 Nello Cristiani and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, eleventh edition, 2010. 41

CT10 Caroline Chibelushi and Mike Thelwall. *Text mining decision elements from meeting transcripts*, volume 52 of *Lecture Notes in Electrical Engineering*. Springer, Heidelberg, 2010. 7, 29

CWhM01 Freddy Y. Y. Choi, Peter Wiemer-hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings of EMNLP*, Pittsburgh, 2001. 45

DF99 S. Dharanipragada and Martin Franz. Story segmentation and topic detection in the broadcast news domain. In *Proceedings of the DARPA Broadcast News Workshop*, 1999. 42, 45, 54

Dha00 S. Dharanipragada. Story segmentation and topic detection in the broadcast news domain. In *DARPA Broadcast News Workshop*, 2000. 43

Dij59 E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 34

DR07 A. Dielmann and S. Renals. Automatic meeting segmentation using dynamic Bayesian networks. *IEEE Transactions on Multimedia*, 9:25 – 36, 2007. 7, 12, 47

Eis09 Jacob Eisenstein. Hierarchical text segmentation from multi-scale lexical cohesion. *The 2009 Annual Conference of the North American Chapter of the ACL*, pages 353–361, 2009. 8

Fow02 Martin Fowler. *Patterns of enterprise application architecture*. Addison-Wesley Professional, Toronto, 2002. 13

FRWP03 Martin Franz, Bhuvana Ramabhadran, Todd Ward, and Michael Picheny. Automated transcription and topic segmentation of large spoken archives. In *Eurospeech 2003*, Geneva, Schweiz, 2003. 29, 39

Ful08 Marguerite Fuller. Using term clouds to represent segment-level semantic content of podcasts. Technical report, The Johns Hopkins University, Baltimore, Maryland, 2008. 26

Gan07 John F. Gantz. The expanding digital universe. Technical report, International Data Corporation, 2007. 1

GCA06 Maria Georgescul, Alexander Clark, and Susan Armstrong. Word distributions for thematic segmentation in a support vector machine approach. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, New York, 2006. 19, 29, 31, 41, 45, 46, 50

GGS10 Camille Guinaudeau, Guillaume Gravier, and Pascale Sébillot. Improving ASR-based topic segmentation of TV programs with confidence measures and semantic relations. In *Proceedings of INTERSPEECH 2010*, pages 1365–1368, Chiba, 2010. 34

GHJV94 Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns*. Addison-Wesley Professional, Toronto, 1994. 66, 67

Giv83 Talmy Givón. *Topic continuity in discourse: a quantitative cross-language study.* Arshdeep Singh, Netherlands, 1983. 5

GK03 Mark Girolami and A. Kaban. On an equivalence between PLSI and LDA. In *Proceedings of SIGIR 2003*, New York, 2003. 42

GMFLJ03 Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 562 – 569, Sapporo, 2003. 28, 45, 46, 50, 54, 55

GNP05 A. Gruenstein, J. Niekrasz, and M. Purver. Meeting structure annotation: Data and tools. In *6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, 2005. 8

GPHJ09 Anuj Goyal, P. Punitha, Frank Hopfgartner, and Joemon M. Jose. *Split and merge based story segmentation in news videos*, volume 5478/2009 of *Advances in Information Retrieval*, pages 766–770. Springer, Heidelberg, 2009. 12, 47

GRZW07 Amit Gruber, Michal Rosen-Zvi, and Yair Weiss. Hidden topic markov models. In *AISTATS*, San Juan, 2007. 40

GS86 B.J. Grosz and C.L. Sidner. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12, 1986. 10, 47

GZ09 David Gibbon and Liu Zhu. *Introduction to Video Search Engines.* Springer, Berlin, 2009. 73

Hea97 Marti A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23:33–64, 1997. 7, 25, 26, 29, 30, 50, 51

Hei98 Oskari Heinonen. Optimal multi-paragraph text segmentation by dynamic programming. In *ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, 1998. 25

HH76 Ruqaiya Hasan and Michael Halliday. *Cohesion in English*. Longman Group, New York, 1976. 5, 24

HM91 Graeme Hirst and Jane Morris. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17, 1991. 27

Hof99 Thomas Hoffmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, Berkeley, 1999. 42

IMKS04 Ichiro Ide, Hiroshi Mo, Norio Katayama, and Shin'ichi Satoh. Topic threading for structuring a large-scale news video archive. *Lecture Notes in Computer Science*, 3115/2004, 2004. 27

JK95 J.S. Justeson and S.M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1, 1995. 11

JZ03 Xiang Ji and Hongyuan Zha. Domain-independent text segmentation using anisotropic diffusion and dynamic programming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, Toronto, 2003. 30, 32, 33, 36, 45, 50, 72

Kat96 S.M. Katz. Distribution of context words and phrases in text and language modeling. *Natural language Engineering*, 2, 1996. 29

Kau99 Stefan Kaufmann. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th annual meeting of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Stanford, 1999. 26

KF94 Hideki Kozima and Teiji Furugori. Segmenting narrative text into coherent scenes. *Literary and Linguistic Computing*, 9, 1994. 24

KG09 Roman Kern and Michael Granitzer. Efficient linear text segmentation based on information retrieval techniques. In *International Conference on Management of Emergent Digital EcoSystems*, Lyon, 2009. 27, 45, 50, 51, 52

KIO96 Jiro Kiyama, Yoshiaki Itoht, and Ryuichi Oka. Automatic detection of topic boundaries and keywords in arbitrary speech using incremental reference interval-free continuous DP. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1946 – 1949, Philadelphia, 1996. 12, 47

KL10 Jáchym Kolář and Yang Liu. Automatic sentence boundary detection in conversational speech: A cross-lingual evaluation on English and Czech. In *Proceedings ICASSP 2010*, 2010. 23, 72

Koz93 Hideki Kozima. Text segmentation based on similarity between words. In *31st annual meeting on Association for Computational Linguistics*, pages 286 – 288, Ohio State University, 1993. 7, 24, 25

KPRS09 Krishna Kummamuru, Deepak Padmanabhan, Shourya Roy, and L. Venkata Subramaniam. *Unsupervised segmentation of conversational transcripts.* Wiley Periodicals, Inc., 2009. 5

Li76 Charles N. Li. *Subject and topic: a new typology of language.* Academic Press, New York, 1976. 5

LJCC04 Ming Lin, Jay F. Nunamaker Jr., Hsinchun Chen, and Michael Chau. Segmentation of lecture videos based on text: A method combining multiple linguistic features (pdf). In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Hawai, 2004. 26

LLS95 Nils Lenke, Hans-Dieter Lutz, and Michael Sprenger. *Grundlagen sprachlicher Kommunikation.* Wilhelm Fink Verlag, München, 1995. 9

LP95 Diane J. Litman and Rebecca J. Passonneau. Combining multiple knowledge sources for discourse segmentation. In *33rd annual meeting on Association for Computational Linguistics*, pages 108 – 115, Cambridge, Massachusetts, 1995. 42

LP07 Alexandre Labadié and Violaine Prince. *Text segmentation based on document understanding for information retrieval*, volume 4592 of *Lecture Notes in Computer Science - Natural Language Processing and Information Systems*, pages 295–304. Springer, Heidelberg, 2007. 11, 27

LP08 Alexandre Labadié and Violaine Prince. The impact of corpus quality and type on topic based text segmentation evaluation. In *International Multiconference on Computer Science and Information Technology*, Wisla, 2008. 20, 46

LTM10 Mihaiela Lupea, Doina Tatar, and Zsuzsana Marian. Learning taxonomy for text segmentation by formal concept analysis. *Proceedings of Symbolic and Numeric Algorithms for Scientific Computing*, 2010. 7, 31

LV00 Peter Lyman and Hal R. Varian. How much information? `http://www2.sims.berkeley.edu/research/projects/how-much-info/broadcast.html`, 2000. [Online; accessed 19.10.2010]. 1, 7

May98 Mark T. Maybury. Discourse cues for broadcast news segmentation. In *International Conference On Computational Linguistics archive Proceedings of the 17th international conference on Computational linguistics*, Montreal, 1998. 12, 13, 47

MB07 Igor Malioutov and Regina Barzilay. Minimum cut model for spoken lecture segmentation. Technical report, MIT - Computer Science and Artificial Intelligence Laboratory, 2007. 31, 34, 35, 36, 45

MH06 J. D. Moore and P. Y. Hsueh. Automatic topic segmentation and labeling in multiparty dialogue. *Spoken Language Technology Workshop*, 2006. 42

MHG+10 Hemant Misra, Frank Hopfgartner, Anuj Goyal, P. Punitha, and Joemon M. Jose. Tv news story segmentation based on semantic coherence and content similarity. *Lecture Notes in Computer Science*, 5916/2010(347/357), 2010. 47

Mit99 Ruslan Mitkov. *Anaphora resolution: the state of the art.* University of Wolverhampton, Wolverhampton, 1999. 7

Mit02 Ruslan Mitkov. *Anaphora resolution.* Pearson ESL, 2002. 7

Mit06 Tom Mitchell. The discipline of machine learning. Technical report, Carnegie Mellon University, Pittsburgh, 2006. 36

MKSW99 John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, Herndon, 1999. 16

ML07 Irina Matveeva and Gina-Anne Levow. Topic segmentation with hybrid document indexing. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007. 42, 45

MMW09 Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. A new quality measure for topic segmentation of text and speech. In *Conference of the International Speech*, Brighton, 2009. 19

MMW10 Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. Discriminative topic segmentation of text and speech. In *13th International Conference on Artificial Intelligence and Statistics*, Sardinia, 2010. 40, 41, 45

MPBG07 Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. Making sense of sound: Unsupervised topic segmentation over acoustic input. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, page 504–511, 2007. 12, 47

MW98 Jim Miller and Regina Weinert. *Spontaneous spoken language: syntax and discourse*. Oxford University Press, Oxford, 1998. 8

MYJC09 Hemant Misra, François Yvon, Joemon M. Jose, and Olivier Cappé. Text segmentation via topic modeling: An analytical study. In *Proceeding of the 18th ACM conference on Information and knowledge management*, Hong Kong, 2009. 36, 45, 47, 50, 72

NMPM03 H. Ney, C. Meyer, J. Peters, and E. Matusov. Topic segmentation using markov models on section level. *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 471 – 476, 2003. 39, 40, 45, 46

NN94 Tadashi Nomoto and Yoshihiko Nitta. A grammatico-statistical approach to discourse partitioning. In *Proceedings of the 15th conference on Computational linguistics*, volume 2, Kyoto, 1994. 5, 24

PC95 J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21 Annual Conference on Research and Development in Information Retrieval*, Washington, 1995. 9

PC97 Jay M. Ponte and Bruce Croft. Text segmentation by topic. In *Lecture Notes in Computer Science*, volume 48. Springer, Heidelberg, 1997. 35

PH02 Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36, 2002. 18

PI10 Raji R. Pillai and Sumam Mary Idicula. Linear text segmentation using classification techniques. In *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, Peelamedu, 2010. ACM. 39, 42

Pou09 Gert-Jan Poulisse. News story segmentation in multiple modalities. In *Multimedia Tools and Applications*. Springer, Heidelberg, 2009. 12, 47

PP10 Raúl Abella Pérez and José Eladio Medina Pagola. An incremental text segmentation by clustering cohesion. In *Proceedings of the first international workshop on handling concept drift in adaptive information systems: importance, challenges and solutions*, Barcelona, 2010. 32, 45, 50

PPK04 Fragkou Pavlina, Vassilios Petridis, and Athanasios Kehagias. A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information Systems*, 23:2:179–197, 2004. 30, 33, 36, 37, 45

PS83 Livia Polanyi and Remko J. H. Scha. On the recursive structure of discourse. In *Connectedness in sentence, discourse and text: Proceedings of the Tilburg conference held on 25 and 26 January 1982.*, Tilburg, 1983. 5, 8

Pur11 Matthew Purver. *Spoken language understanding: systems for extracting semantic information from speech*, chapter Topic Segmentation. John Wiley & Sons, Ltd, 2011. 6, 8, 17, 73

Rey98 Jeffrey C. Reynar. *Topic segmentation - algorithms and applications.* PhD thesis, University of Pennsylvania, Pennsylvania, 1998. 5, 6, 8, 10, 13, 30, 31, 32, 73

Rey99 Jeffrey C. Reynar. Statistical models for topic segmentation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Maryland, 1999. 39

Rie02 Klaus Ries. *Segmenting conversations by topic, initiative, and style*, volume 2273 of *Lecture Notes in Computer Science.* Springer, Heidelberg, 2002. 43

RR97 Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *ANLC '97 Proceedings of the fifth conference on Applied natural language processing*, Philadelphia, 1997. University of Pennsylvania. 72

SB07 Laurianne Sitbon and Patrice Bellot. Topic segmentation using weighted lexical links (WLL). In *SIGIR 2007 Proceedings*, 2007. 28, 29

SCS02 Nicola Stokes, Joe Carthy, and Alan F. Smeaton. Segmenting broadcast news streams using lexical chains. In *1st Starting AI Researchers Symposium*, 2002. 27, 28

SCS04 Nicola Stokes, Joe Carthy, and Alan F. Smeaton. Select: A lexical cohesion based news story segmentation system. *AI Communications*, 2004. 27, 28

SL08 M. Sherman and Yang Liu. Using hidden markov models for topic segmentation of meeting transcripts. *Spoken Language Technology Workshop*, pages 185 – 188, 2008. 29, 39, 46, 50

SLLW08 Qi Sun, Runxin Li, Dingsheng Luo, and Xihong Wu. Text segmentation with LDA-based Fisher kernel. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, 2008. 35, 36, 45, 50

SM08 M. Mahdi Shafiei and Evangelos E. Milios. A statistical model for topic segmentation and clustering. *Lecture Notes in Computer Science*, 5032/2008:283–295, 2008. 7, 8, 38

SSHTT00 Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, pages 127–154, 2000. 12, 47

Ste10 Steffikey. Figure to the Wikipedia article Support Vector Machine. http://de.wikipedia.org/w/index.php?title=Datei:Diskriminanzfunktion.png, 2010. [Online; accessed 02.12.2010]. 41

UI01 Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001. 33, 34, 45, 46, 50

Way00 Charles L. Wayne. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Language resources and evaluation conference*, Athen, 2000. 20

Web08 Merriam Webster. *Webster's new college dictionary*, page 1190. Houghton Mifflin Harcourt, Third New Updated edition, 2008. 4, 5

Wei09 Eugene Weinstein. *Search Problems for Speech and Audio Sequences*. PhD thesis, Department of Computer Science Courant Institute of Mathematical Sciences, New York University, 2009. 73

Wu09 Chung-Hsien Wu. Story segmentation and topic classification of broadcast news via a topic-based segmental model and a genetic algorithm. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 8, pages 1612–1623, 2009. 43

Yaa97 Yakov Yaari. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of the RANLP'97*, Borovets, 1997. 30

You91 Gilbert Youmans. A new tool for discourse analysis: the vocabulary-management profile. *Language*, 67.4, 1991. 10, 24

Yul96 George Yule. *The study of language*. Cambridge University Press, Cambridge, 1996. 5, 6

YvMLG98 J. P. Yamron, P. van Mulbregt, S. Lowe, and L. Gillick. A hidden markov model approach to text segmentation and event tracking. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*, Seattle, 1998. 39, 45

YZZ+08 Na Ye, Jingbo Zhu, Yan Zheng, Matthew Y. Ma, Huizhen Wang, and Bin Zhang. A dynamic programming model for text segmentation based on min-max similarity. *Proceedings of the 4th Asia information retrieval conference on Information retrieval technology*, 2008. 36, 37

A. MPEG7/XML example (a fragment)

Listing A.1: *MPEG7 news transcript fragment. The StartTimeDurationMatrix node defines the transcript timecodes. The transcript itself is contained in the SpokenUnitVector node.*

```
1 <ns2:Transcription linguisticUnit="word" mediaTimeBase="../../MediaTime[1]/MediaTimePoint
      " mediaTimeUnit="PT1N1000F">
2   <ns2:StartTimeDurationMatrix ns1:dim="24 2">
3     101720 1910 103640 90 103740 360 104110 450 104570 170 104750 830 105590 180 105780 140
          105930 520 106460 1590 108060 310 108380 440 108830 580 109420 110 109540 200
          109750 570 110330 150 110490 340 110840 200 111050 280 111340 370 111720 570
          112300 180 112490 630
4   </ns2:StartTimeDurationMatrix>
5   <ns2:ConfidenceVector>0.97697794 0.70402163 0.9604682 0.99597824 0.96934783 0.94806993
        0.91987723 0.54550016 0.9948822 0.06486012 0.9232542 0.6841647 0.40412095 0.33167678
        0.99604803 0.2071835 0.98148304 0.99634176 0.9708524 0.86709917 0.0039638006
        0.6995697 0.43390822 1.0</ns2:ConfidenceVector>
6   <ns2:SpokenUnitVector>
7     sil die letzten Umfragen haben gezeigt dass ein SPD Buergermeisterkandidat Olaf Scholz
          sowohl an von Beust und auf alle Faelle Handal ausschlagen kann sil
8   </ns2:SpokenUnitVector>
9 </ns2:Transcription>
```

B. Best-performing topic segmentation methods

	A	B	C	D	E	F	G	H	I
1	**Year**	**BibTeX**	**Author**	**ASR**	**P_k**	**Training**	**Methods**	**Best-perf. corpus**	**Comment**
2	2009	MYJC09	Misra et al.	no	**2%**	yes	LDA, graph analysis	Choi trained on RCV1	adapted value; unadapted up to 20%
3	2003	JZ03	Ji et al.	no	**4%**	no	dotplotting, anisotropic diff.	Choi 9-11 (avg=5%)	
4	2004	PPK04	Fragkou et al.	no	**4%**	yes	DP, dotplotting	Choi	
5	2008	SLLW08	Sun et al.	no	**5%**	no	LDA, DP	Choi-like	Chinese web news
6	2009	KG09	Kern/Granitzer	no	**5%**	no	TextTiling derivate	RCV1	
7	2006	Bes06	Bestgen / Choi	no	**6%**	yes	LSA dotplotting, ranking	Choi, Choi-like	
8	2002	BCT02	Brants et al.	no	**8%**	yes	PLSA	RCV1, Choi	
9	2001	UI01	Utiyama	no	**10%**	no	DP/stat. model	Choi	ASR=Pk=35.2% (33 physics lectures WER=19%)
10	2010	PP10	Peréz et al.	no	**10%**	no	overlapping clusters	scientific papers	
11									
12									
13	2003	GMFLJ03	Galley et al.	yes	**7%**	no	lexical chains	TDT	TDT, 32% on ICSI
14	2003	NMPM03	Matusov et al.	yes	**9%**	yes	section-level HMM	medical reports	
15	1997	BBL97	Beeferman et al.	yes	**12%**	yes	exponential models	TDT	
16	1999	DF99	Dharanipragada	yes	**16%**	yes	decision tree	WSJ, BN	WSJ: 38M words, BN: 150M newsarticles
17	2007	ML07	Matveeva/Levow	yes	**17%**	no	GLSA	TDT	
18	1998	YvMLG98	Yamron et al.	yes	**18%**	yes	HMM, speech recognizer	TDT	
19	2000	Cho00	Choi	yes	**21%**	no	dotploting with ranking	Choi	
20	2006	GCA06	Georgescul et al.	yes	**22%**	yes	SVM	ICSI	Brown 19%, TDT 20%
21	2010	MMW2010	Mohri, Weinstein	yes	**24%**	yes	SVM	TDT	

Table B.1.: *Details on the best-performing topic segmentation methods separated in ASR-tested (top) and other methods (bottom). Legend: unspervised methods are highlighted in gray.*

C. Online topic annotation tool

Figure C.1.: *Online topic annotation tool, allowing persistent segmentation sessions for multiple users. Inerface details: a) video player; b) timeline showing saved segment markers; c) asset list presented to the user; d) current timestamp; e) list of topic boundaries created by the current user.*

The tool is available on the web: http://topics.jls-hosting.net.

D. A configuration example for topic segmentation execution

Listing D.1: *A configuration example for the bean factory defining a topic segmentation executions. Algorithm selection and configuration is followed by a set of filters and error metric modules.*

```
<configuration
    name="lcseg"
    language="ger"
    outputPath="c:/tsout/tagesschau-lcseg"
    report="yes"
    reference="testdata/topicsegmentation/tagesschau/reference.txt"
    stopwords="resource/stopwords/stopwords_de.txt"
    extendedCharting="false"
    chartWidth="800" chartHeight="300"
    idf ="testdata/topicsegmentation/tagesschau/corpus.idf"
    >

    <description>LCSeg Galley et al. (2007)</description>

    <!--
    <transcriptProvider class="de.fhg.iais.aglu.topicseg.transcript.AMTranscriptProvider">
      <param type="string" name="wsdlLocation" value="http://localhost:8080/audiomining/
          mediaArchive?wsdl" />
      <param type="string" name="index" value="tagesschau" />
    </transcriptProvider>
     -->

    <transcriptProvider class="de.fhg.iais.aglu.topicseg.transcript.MPEG7TranscriptProvider"
        >
      <param type="string" name="corpusFolder" value="testdata/topicsegmentation/
          tagesschau"></param>
    </transcriptProvider>

    <segmenter class="de.fhg.iais.aglu.topicseg.TopicSegmenter02">
      <param type="int" name="blockSize" value="2" />
      <param type="int" name="sentenceLength" value="16" />
      <param type="int" name="hiatusThreshold" value="176" />
      <param type="int" name="chainableThreshold" value="2"/>
      <param type="boolean" name="charting" value="true"/>
    </segmenter>
```



```
33
34    <filters>
35      <filter class="de.fhg.iais.aglu.topicseg.filter.MovingAverage" name="movingaverage"
            enabled="true">
36        <param type="int" name="windowSize" value="3" />
37      </filter>
38      <filter class="de.fhg.iais.aglu.topicseg.filter.LocalMinima" name="localminima"
            enabled="true">
39      </filter>
40      <filter class="de.fhg.iais.aglu.topicseg.filter.MetaFilter" name="metafiler" enabled="
            true">
41        <param type="int" name="silenceThreshold" value="1200"/>
42        <param type="int" name="searchRadius" value="20"/>
43      </filter>
44      <filter class="de.fhg.iais.aglu.topicseg.filter.SmoothingNeighbours" name="
            smoothneighbours" enabled="true">
45        <param type="double" name="plimit" value="1" />
46        <param type="double" name="alpha" value="0.5"/>
47        <param type="boolean" name="charting" value="true"/>
48        <param type="string" name="color" value="lightgray"/>
49      </filter>
50
51    </filters>
52
53    <metrics>
54      <metric class="de.fhg.iais.aglu.evaluation.topicseg.BeefermanCalculator" name="pk"
            enabled="true" >
55      </metric>
56      <metric class="de.fhg.iais.aglu.evaluation.topicseg.WDCalculator" name="wd" enabled=
            "true" >
57      </metric>
58      <metric class="de.fhg.iais.aglu.evaluation.topicseg.PRThresholdCalculator" name="prt"
            enabled="true">
59        <param type="int" name="windowSize" value="50" />
60      </metric>
61    </metrics>
62
63  </configuration>
```

Index

abstract factory, 66
anaphora, 5, 7
artificial corpora, 20

Bayesian network, 39
Beeferman error metric, 17
bigram, 11

cohesive links, 5
content recommendation, 7
conventional corpora, 20
corpus, 20
cosine measure, 23, 31, 51
cue words and phrases, 10

decision analysis, 7
depth score, 26
directed acyclic graph, 39
discourse, 5
Dotplotting topics, 31

event, 6

F-score is, 16
first uses, 10
Fisher kernel, 36
formal concept analysis, 31

genetic algorithm, 43
gold standard, 15

hard data, 8
hiatus, 54
hierarchical segmentation, 8
homographs, 11
hyperplane, 40

indicators of topical structure, 9
inflexions, 10

k-means, 32
kernel functions, 41

language diversity, 9
language model, 9
latent Dirichlet allocation, 35
latent variables, 38
lexical chains, 27, 50, 55
lexical cohesion, 5
local context analysis, 35

Markov chain, 39
maximum entropy, 39

n-gram, 11
negative clue, 9, 11
normalized-cut criterion, 35

optimization problem, 32
oversegmentation, 16

paraphrasing, 9
perception, 8
pointwise mutual information, 20
precision, 16
prior probability distribution, 36

query expansion, 35

recall, 16
recursion, 5

self-nomalized segmentation, 15
semantic network, 24, 25

semantic proximities, 35
sentence boundaries, 72
sentences in speech, 8
single value decomposition, 41
speech event, 6
spontaneous speech, 48
spreading activation, 24
stemming algorithm, 10
stochastic process, 38
stop word removal, 11
subtopics, 8

text complexity, 11
text summarization, 7
topic particle, 5
topic shifts, 6
topic-prominent, 5
transcript preprocessing, 10
tropes, 9

unigram, 11

