Rheinisch-Westfälische Technische Hochschule Aachen
Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme
Rheinische Friedrich-Wilhelms-Universität Bonn
B-IT Center for Information Technology

A Semantic Content-based Recommender System using Bayesian Networks

Master's thesis
in Media Informatics

submitted by
Aya Kamel
Matr. Nr. 329043

Supervisor:
Dr. Rüdiger Klein
Adaptive Reflective Teams Group, Fraunhofer-Institut IAIS

First Examiner:
Prof. Dr.-Ing. Christian Bauckhage
Visual & Social Media Group, Fraunhofer-Institut IAIS
B-IT Center for Information Technology

Second Examiner:
Prof. Dr. Sören Auer
Organized Knowledge Group, Fraunhofer-Institut IAIS
Institut für Informatik III, Universität Bonn

I hereby affirm that this work was created exclusively by myself and that no other sources and auxiliary means other than the ones stated were used. All material or text which was extracted literally or analogously from other published work is explicitly stated.

October 27, 2014

Aya Kamel

Acknowledgements

First of all, I would like to thank God for giving me the might and power to work on this project. I owe my deepest gratitude to my supervisor Dr. Rüdiger Klein for his continuous support, valuable guidance and encouraging trust. I am really honored to have worked under his supervision and I definitely learnt a lot from him. No words can express my extreme appreciation and gratitude to my professor and mentor Prof. Dr. Christian Bauckhage who not only enriched my academic knowledge but also taught me how to be an independent thinker. Special thanks to my leader Jinguan Xie for the valuable discussions and the friendly stress-free work environment he offered me. I would also like to thank Prof. Dr. Sören Auer for the honor of reviewing my work in this project. I am indebted to my awesome friends who stood by my side throughout this phase of my life and were a permanent source of inspiration, motivation and support. Finally and most importantly, I would like to express my deep and special thanks to my one-of-a-kind family who helped me be who I am today and never failed to support and encourage me even while being thousands of miles away.

Abstract

In the era of digital world and WWW, most of the human activities have slowly started to be tightly coupled to the Internet. Like all other forms of multimedia web content, the amount of video content on the web has increased drastically over the past decade, reinforcing the need for Recommender Systems to help users reach relevant and interesting content. In an attempt to extend the research in the field of Recommender Systems by introducing cutting edge technologies, this thesis proposes a new recommendation approach in which Bayesian Networks are used for semantic aware reasoning about users interests. The theoretical proposal is accompanied by an illustrative implementation which is evaluated to verify the applicability of this approach. The results show that the proposed approach shows a very promising potential in practical applications with some limitations that could be further investigated for improvement.

Contents

List of Figures

Chapter 1

Introduction

In the era of digital world and WWW, most of the human activities have slowly started to be tightly coupled to the Internet. Nowadays, most of the people watch TV, read books and even do their shopping online. Recommender Systems (RSs) are software engines which help Internet users find items like products, articles, videos or books that fit their interests by providing suggestions [54]. With the massive increase in the amount of products and web content, Internet users have started to heavily rely on RSs to reach whatever items they are looking for. Furthermore, the RSs became to some extent in control of what information or products the user will be able to find, since it is becoming the only window for users to reach such items. Realizing the importance of the role played by the RSs, companies have decided to invest in the research of RSs as their future marketing window.

1.1 Motivation

Like all other forms of web content, the amount of multimedia content on the web has increased drastically over the past decade. For example, according to a recent study, the US internet users watched around 49 billion videos online in January 2014 alone [40]. Facing this huge amount of multimedia content, the users face the question "What should I watch/read next?". Efficient multimedia recommendation technologies serving the purposes of information surfing or entertainment answer this question. Multimedia item recommendations would direct the users towards other items that they have not yet watched or read, and that could be of interest to them. For example for a user who likes football, interesting recommendations could be videos showing the best trending football matches that this user has not yet watched or the latest news articles about this user's favorite team. Several RSs were developed to address the problem of suggesting relevant multimedia items.

Studying the behavior of current RSs, it was noticed that very few of them actually consider the content of the item, or try to understand what the user's interests really are. Instead, they focus on finding the users with the closest feedback history, and recommend

whatever items these users liked regardless of what the contents of the items are. For example, if two users A and B like some common videos, it is assumed that A and B are similar users and therefore, whatever other videos that A liked are recommended to B regardless of what these other videos show. The problem with this approach is that the similarity assumption may not be very accurate. For example it can happen that, despite that A and B both liked some common videos, A's interests are football and politics in general. Whereas, B is only interested in German football and is not at all interested in politics. The only way to realize this difference is to dig deeper into the contents of the videos that each user liked, and try to find the patterns which best describe the interests of each user individually.

As a workaround to this limitation, another direction in RSs development emerged in which each item is represented as a set of annotations, which approximate the contents of this item. Items are then recommended based on their contents rather than on user history similarities. For example a video showing a match between FC Bayern Munich versus FC Barcelona in the UEFA League would be annotated with the set of tags FC Bayern Munich, FC Barcelona, UEFA. It was noticed however that the implementations of the RSs following this approach use the annotations only as keywords and convert the problem into a trivial keyword search, losing information about the semantics of the content. For example, the information that FC Bayern Munich and FC Barcelona are German and Spanish teams respectively is lost. In that manner, it is not possible to detect specific interest patterns. For example, for a user who watches only videos showing matches involving FC Bayern Munich, BVB Dortumnd and Schalke, it is not possible to realize the pattern that they are German football teams. As a result to this missing link caused by ignoring semantics, the user's interest cannot be accurately inferred. Most recently, some RSs started considering semantics in their recommendations. However, they use a set of complex static heuristic rules which require expert knowledge to develop.

1.2 Objective

With the emergence of Semantic Web technologies, the information on the WWW is becoming more structured, and related information about almost anything can be relatively easily fetched from Linked Open Data (LoD) Knowledge Bases. This consequently opens the door for utilizing such technologies in enhancing and extending applications that involve human reasoning simulation tasks, like the one addressed in this thesis. On the other side, the development of system intelligence is rapidly switching from the era of soft coded statements, where the developers transform their knowledge into a set of static rules, towards the era of dynamic learning, where the machines independently learn such rules. Studying the nature of the RSs problem, it was found that it resembles machine learning problems where given some training user feedback about some items, the system is required to predict the user feedback for unobserved items. It was also found that the nature of this problem is full of uncertainty, whether in terms of feedback confidence or annotations accuracy.

This thesis proposes, exploits and evaluates a new approach for detecting interest patterns for users, through content-based analysis of the users' watch histories in order to learn the user preferences, and accordingly generate multimedia item recommendations, with specific focus on video recommendations. The proposed approach overcomes the limitations and shortcomings of the state of the art RSs, by merging the advanced technologies of semantic web and machine learning into a new system. The system models semantic relations between video contents in the form of a Bayesian Network (BN). The BN in turn is trained through evidence propagation to find interest peaks in the users watch history, utilizing the ability of BNs to handle probabilistic uncertainty. The work towards this thesis was performed as part of the EU project LinkedTV [1], whose goal is to personalize the TV experience for users, not only by directing them to content that is of interest to them, but also by smartly enriching their experience through attaching relevant information about the content played to a second screen.

This thesis is divided into 6 chapters. Chapter 2, introduces each of the three corners of this approach; RS, Semantics and Bayesian Networks, following that with a more detailed study of the related work in this field and a more generous motivation for the new approach based on the introduced knowledge. Afterwards in Chapter 3, the workflow in the tightly coupled project LinkedTV is presented, highlighting the older system's limitations and theoretically elaborating the proposed approach. Chapter 4 gives an overview of the implementation details of the new RS that uses the previously proposed approach. Following that, the means and results of the system evaluation are presented in Chapter 5. Finally, Chapter 6 concludes the thesis by highlighting the positive findings as well as the challenges of the experiments performed throughout this project and offers suggestions for possible future work.

[1] www.linkedtv.eu

THIS PAGE IS INTENTIONALLY LEFT BLANK

Chapter 2

Background

This chapter is divided mainly into four sections. Section 2.1 sheds light on Recommender Systems and the different approaches and techniques developed for this purpose. Section 2.2 addresses the topic of Semantic Web and its underlying concepts. In Section 2.3, the key concepts of Bayesian Networks and inference algorithms are presented. Finally, Section 2.4 reviews, given the knowledge presented in the previous sections, the currently available Recommender Systems and justifies the need for a new system.

2.1 Recommender Systems

Nowadays in the everyday life, people heavily rely, even without them explicitly noticing, on RSs to guide them through the massive amount of multimedia content available on the Internet be it YouTube to watch videos, Amazon to buy products or Netflix to watch movies. The first standalone RS was presented in a Patent by J.B. Hey in the early 1990s [34]. Ever since, RSs have joined the trendy research areas which until these days have not lost its position as a hot research topic due to its great role regarding helping users through the massive amount of information [2].

A RS is simply a system which given some information about a user within larger groups of users, manages to decide how relevant certain contents could be for this user and accordingly generate recommendations to the user about which content to consider [54]. This problem can be mathematically formulated as follows: Given a set of users U in which each user is defined by a *profile* and a set of content items C in which each item is defined by some *content characteristics*, the task is to first compute the relevance function f of each content in C to each user in U, $f : U \times C \rightarrow F$ where F is an ordered set of real numbers within a certain range. Then the best item $c' \in C$ is generated as

$$\forall u \in U \ , \ c'_u = \arg\max_{c \in C} f(u, c)$$

The user profile describes the user either simply through holding personal information about the user like age, gender, ...etc or in a more fine grained manner through information

regarding how much this user rated some other content previously. On the other hand, the content characteristics describe the content itself usually by the meta-data supported by the content author [2].

There is a variety of approaches to generate recommendations. The following section describes some of these approaches.

2.1.1 Recommendation Approaches

As explained above, the main functionality of a RS is to match the user's interests with the available multimedia content and accordingly recommend to the users what would be relevant for them. The process of computing the relevance function for some content to some user differs according to the approach being used. In general, RSs can be classified into 3 major classes based on the recommendation approach; collaborative, content-based and hybrid RSs. [2].

Collaborative Recommendation

The early RSs that emerged in the mid 1990s like Tapestry [29] and GroupLens [53], used the collaborative recommendation (also referred to as collaborative filtering) approach. The general idea in this approach is to direct the user towards content that other users with similar interests were found to like. The recommendation problem then becomes, how to find the users which have similar interests as the user currently considered, given the ratings these users gave to other content.

There are two methods to generate collaborative recommendations; model-based and neighborhood methods [54]. In model-based methods, the idea is to use the ratings provided by users to train a model of user classes and categories of items, that is, to cluster users based on the ratings they give to the contents, and then later use this model to predict new weights. The neighborhood methods, on the other hand, use the given ratings directly to compute similarity values, like for example, the root mean square of the difference between the ratings of the same items.

Collaborative filtering has the advantage that it does not require descriptors to the content items because the recommendations are solely generated based on the ratings given by other users. It however fails to generate recommendations for content that has not been rated by the "similar" users [54]. Many of the famous recommendation systems like Youtube[20] and Amazon[41] use this approach for recommendations. A more detailed discussion of these systems will follow in Section 2.4.1.

Content-based Recommendation

Content-based recommendations which are the focus of this thesis first appeared in the mid 1990s in systems like Newsweeder [38]. The general idea in this approach is to find

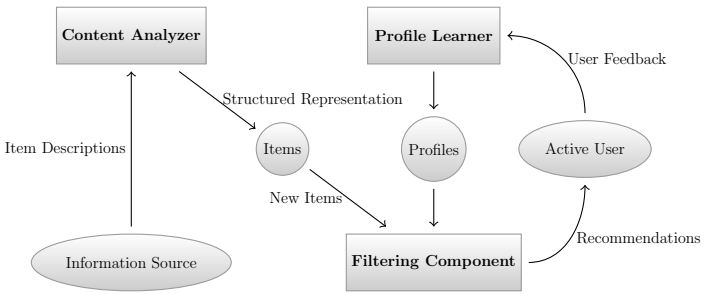


Figure 2.1: The high level data flow in a content-based RS [54]

out what is the common content between the items in which the user previously showed interest and recommend other items which contain the same content.

A basic content-based RS consists of three main modules as shown in Figure 2.1; a content analysis module, an interest learning module and a filtering module [54]. The need for the content analyzer is to extract the contents from the watched items either from meta-data or using some advanced technology to automatically annotate the contents of an item. Up to this day, no suitable methods to accurately represent the video contents are available. Therefore, the content analyzers only generate annotations that approximate the contents of the video by pointing out the main aspects of the video, for example, the people, places, objects and events in the video. The output of this module is then passed to the interest learning module in which a user profile is created with the interesting topics for the user. In the end, the filtering component filters out the items whose contents do not match the learned profile.

Content-based RSs can generate good recommendations in contrast to collaborative RSs in the cases when not many users have overlapping interests and in the situations when no ratings are available for a content item. On the other hand, they still also face the cold start problem like the collaborative filters in case of new users without enough history ratings. In addition, the problem of content analysis is still not solved. As a result, this approach suffers the drawback of being sensitive to bad or sparse analysis data. Content-based RSs in general also suffer from the over-specialization problem [2], which is the problem of restricting the users only to the items that are similar to what they have already showed interest in, without getting any recommendations to other new possible interests.

Hybrid Recommendation

The third class of recommendation approaches is the hybrid recommendation approach which is, as the name implies, a hybrid of the collaborative and content-based approaches. This approach was introduced in systems like Fab [7] in the late 1990s. There are different methods for combining both content-based and collaborative RSs.

One method is to perform collaborative filtering on content-based filtered items. First, a content-based system is used to choose the items whose general topic matches the general interest topics of the user. In this case, the content analysis only needs to be coarse grained on the general topic level, and the same for the interest learning. Then the filtered items go through a collaborative system to re-rank the items based on similar users' feedback. Another method could be to use each system separately and in the end combine the results of both systems using a linear combination function, a voting algorithm, or simply by switching between the results of both systems.

Combining the two previously explained systems avoids some of the limitations imposed by each of them like the problem of the new item in case of collaborative systems or the problem of hard content analysis in case of content-based systems. It still however faces the same cold start problems that the other systems face.

2.1.2 Recommendation Techniques

Another classification of RSs considers the techniques in which the recommendations are generated. From this perspective, RSs can be classified into rule-based systems and model-based systems [2].

In rule-based RSs, also referred to in literature as heuristic systems, a set of rules predefined by some domain experts are used to compute the rankings for the content items. For example, computing the relevance of an item for some user as the sum or average relevance of the same item for a set of similar users.

Contrarily, model-based systems are, as the name implies, based on a model which is trained using some evidence information to draw conclusions or predictions about unobserved items. Examples for model-based RSs are systems which use probabilistic graphical models like Bayesian Networks which will be explained generously later in Section 2.3.

It is hard to argue which technique is better. Rule-based RSs are able to model user interests in a fine-grained manner. However, they have limited reasoning abilities, and they rely on the overhead effort of experts to build the heuristic measures and tweak it to work in practice. On the other hand, the model-based systems are powerful reasoning tools but they do not cover user interests in a fine-grained manner especially in cases of probabilistic models.

2.1.3 Recommendation Considerations

Studying the natural human behavior, researchers have found that the human interests are not simply just static lists of likes and dislikes, but they tend more to be dynamic topic clouds that increase and decrease depending on some variables. In order to give the maximum flexibility and accuracy to RSs, some features need to be added in order to capture these considerations.

One of the considerations in this regard is *context-aware* recommendation. A context in this sense captures the variables that could affect the interests of a user and adds this information into the recommendation models. Examples for variables that could affect users interests include time of the day; one's interests in mornings are not as likely to divert to entertainment purposes as in the evenings, mood; when somebody is in celebration mood they are not likely to watch a drama or generally sad movies, or company; a user's interests with work colleagues differ than those with friends or those with children.

Examples for Context Aware RSs (CARS) are the ones presented by Adomavicius and Tuzhilin in [3] and Abbar et. al in [1] where the general and extremely simple approach is to add a third dimension to the problem. That is, the problem formulation changes from 2.1 to 2.2

$$\text{users} \times \text{content} \rightarrow \text{rating} \tag{2.1}$$

$$\text{users} \times \text{content} \times \text{context} \rightarrow \text{rating} \tag{2.2}$$

Another field of ongoing research to extend the functionality of RSs is generating recommendations for groups (GRS). Most of the RSs focus their attention on improving the quality of recommendations for individuals either through enhancing the models which capture the user interests or through improving the matching functions which clustering users into groups with similar taste. However, sometimes it is needed to exploit the notion of generating recommendations for a group of users together, that is, finding the items whose content would be relevant to all individuals within a group at the same time as a compromise even when this group consists of users of different interests. For example, for a group of two users A and B, where A is interested in Technology and B is interested in Football, a GRS would recommend an item about the new goal detection technologies. One of the leading research projects in this regard is the PolyLens project [46] where the system recommends movies to groups of users.

2.2 Semantics and Semantic Web

In the past few decades, the amount of information on the Internet has increased horrendously. Being a highly available, low cost alternative to books, it has gradually become the main source of knowledge transfer in the world. It's pages hold information about almost anything that can come up on one's mind.

With the massive increase in the amount of information, the efforts that users needed to do in order to find the required information increased accordingly. In an attempt to solve this problem, machine engines currently known as *Search Engines* have been introduced to help users find the pages which may hold the information they are searching for. Users enter the keywords they are looking for, and the engine directs them to the pages in which these keywords exist. This solved a huge part of the problem. However, sometimes users

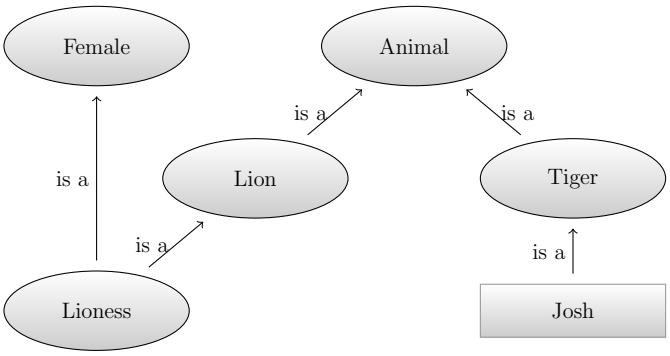


Figure 2.2: Knowledge representation in the form of a Semantic Net

are searching for something more specific, more semantic, where keywords no longer serve their deeds.

At this point, researchers realized that what is really needed is to automatically process the available information on webpages and directly serve answers to the users. But, there they faced further problems; the majority of available information is presented in a natural language, unstructured, human-readable format which machines cannot read, and even if it is machine readable, conclusions cannot be implicitly drawn out of such information because machines lack the ability to do reasoning.

In the late 1990s and early 2000s, Tim Berners Lee, the inventor of the Web, introduced a potential solution; *Semantic Web*. An extension to the current Web back then in which the target is to form a Web of machine understandable networks of connected *data* rather than the existing network of connected *text documents* [13]. The World Wide Web Consortium (W3C) is now organizing the research towards Semantic Web in a worldwide collaborative project. Quoting the W3C in describing Semantic Web, they say: "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries" [16].

2.2.1 How to Represent Semantics?

The notion of semantics and semantic meaning started long before the idea of Semantic Web in the fields of Psychology, Philosophy and Linguistics. The main objective of research in this field was to provide a structural way of representing knowledge and consequently being able to automatically reason about it. In general, there are two types of knowledge; Intensional Knowledge and Extensional Knowledge. Intensional Knowledge is usually the factual knowledge that defines and describes static information about the problem domain. This information usually holds for all instances of time and does not change, for example, tigers are animals. On the other hand, the Extensional Knowledge represents temporary knowledge specific to the problem, for example, Josh is a tiger [6].

One of the earliest structures introduced for the purpose of organizing knowledge is the Semantic Network structure. A Semantic Net as described by Sowa [58] is knowledge represented in the form of a graphical structure which holds patterns of nodes and arcs. Normally nodes would represent concepts and arcs would represent the relations connecting the concepts to each other as shown in Figure 2.2. Having developed a structure for representing knowledge, rose the importance of finding a way to formalize this knowledge. Description Logics (DL) is one of the formalisms developed to represent structural knowledge [6]. In DL, two components were developed to model both types of knowledge explained above; the T-Box and A-Box respectively. The T-Box offers means to declare taxonomic facts either by simple inclusion axioms like

$$\text{Tiger} \sqsubseteq \text{Animal}$$

or by more complex axioms including intersections or unions of concepts like for example

$$\text{Lioness} \equiv \text{Lion} \sqcap \text{Female}$$

In this case, only single definitions of axioms are allowed and no recursion is allowed in the definitions i.e. an axiom cannot be defined by itself [6]. On the other hand, the A-Box offers means to make assertions about individuals like membership to a certain concept for example

$$\text{Tiger (Josh)}$$

Combining T-box declarations with A-Box assertions allows for simple reasoning tasks like for example inferring that if a tiger is an animal and Josh is a tiger then Josh is an animal without explicitly having an axiom stating this information. Nowadays, most of the applications involving artificial intelligence rely in a sense on DL for decision problems. The idea of Semantic Web builds on such concepts, organizes and motivates the use of semantics in applications.

2.2.2 Semantic Web Stack

In order to organize the protocols of communication and to unify the used conventions, the W3C presented the Semantic Web stack of technologies shown in Figure 2.3. The following sections briefly introduce some of the building block technologies in this stack.

RDF

The Resource Description Framework (RDF) [43] is a framework introduced by the W3C to standardize and unify the interchange of data on the Web. It is simply, as the name implies, a language for describing and providing information about the resources available on the World Wide Web.

In RDF, resources, previously referred to as concepts, are represented by Uniform Resource Identifiers (URIs), and are described through properties and property values.

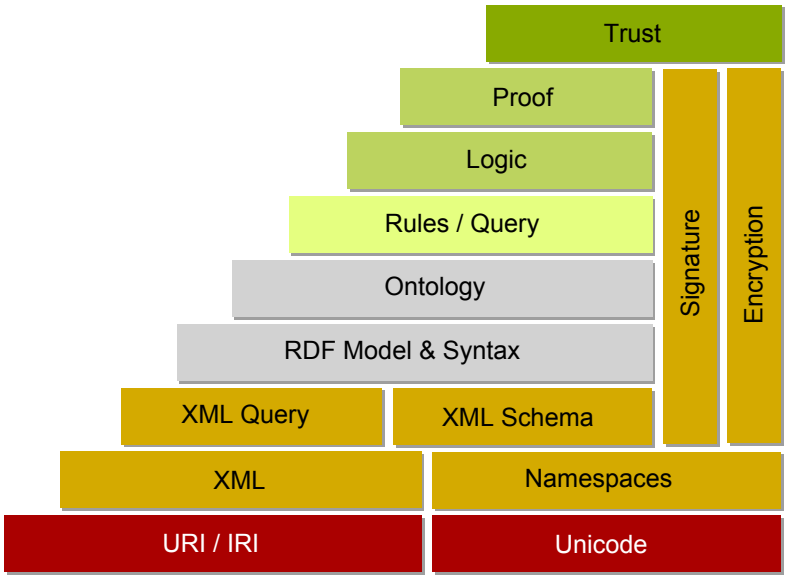


Figure 2.3: The Semantic Web data stack [26]

Property values can either be other resources or simply literals of primitive data types like Integers or Strings. By describing multiple resources, RDF produces graphs of knowledge where nodes are the resources and links are the properties connecting these resources to each other. In this manner, the description of a resource is structured is a way that can be used to easily retrieve useful information. Figure 2.4 shows a simple RDF graph. Notice the resource URIs which define each entity.

In order to save this knowledge in a machine readable format, some formats have been introduced. RDF/XML [28], an XML-based Syntax is one of such formats where RDF graphs are transformed to XML tags. Another format is the Terse RDF Triple Language (Turtle) [9] where the object, property and value triples are directly listed. RDF information can be retrieved using SPARQL [52], the RDF query language.

RDF-S

The RDF Schema (RDF-S) [12] is considered the schematic semantic extension of RDF. In other words, it defines how data modeled in RDF graphs could be structured in a way that is usable for reasoning. RDF-S introduces the notion of a *class* which is a resource grouping a group of resources which share the same properties. The resources that are members in the group are called the *instances* of the class. Classes themselves can be grouped into more general classes. For example, consider a knowledge graph to model scientists. The resources Albert Einstein and Isaac Newton would be instances of the class Scientist, which is a subclass of the more general class Person. RDF-S can support defining this information using properties like `rdfs:subClassOf` to denote the class in-

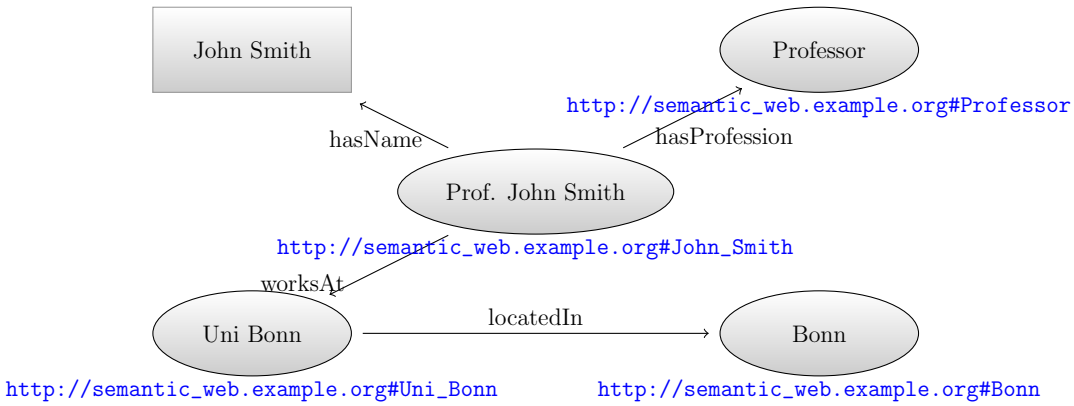


Figure 2.4: Simple RDF Graph

heritance and `rdf:type` to denote membership to a class.

In RDF-S, semantic structuring of the domains and ranges of properties using `rdfs:domain` and `rdfs:range` as well as property inheritance through `rdfs:subPropertyOf` is also possible which adds semantic value to the information inserted. In addition, RDF-S allows differentiating between different types of containers like `rdf:Bag`, `rdf:Seq` and `rdf:Alt` which indicate whether the listed set of resources is unordered, ordered or optional respectively. This is also another form of adding semantics to data.

Following a schema, information can be implicitly deduced from the RDF graph because information is structured in a semantic manner. For example, in the scientists models introduced above, information like "Albert Einstein and Isaac Newton are Persons" can be directly inferred. This means that knowledge does not have to be explicitly stated. However, through reasoning it can be easily retrieved.

OWL

The term Ontology comes from a philosophical origin with the meaning "The science of being". In the field of Computer Science an Ontology is the structural representation of knowledge in a certain domain [30]. The main limitation of the traditional Web, which is addressed through Semantic Web, is the inability of machines to process data and draw conclusions out of it. In Semantic Web and with the help of RDF, data is stored as a graph of interrelated resources whose relations are structured semantically based on a schema (RDF-S). As a result, implicit information could be implicitly retrieved. However, the semantics offered by RDF-S are sometimes not expressive enough to be used for reasoning in real-life applications. Therefore, the W3C introduced the Web Ontology Language (OWL) [44] for extended semantic expressiveness.

There are three different versions of OWL with different levels of expressiveness; OWL

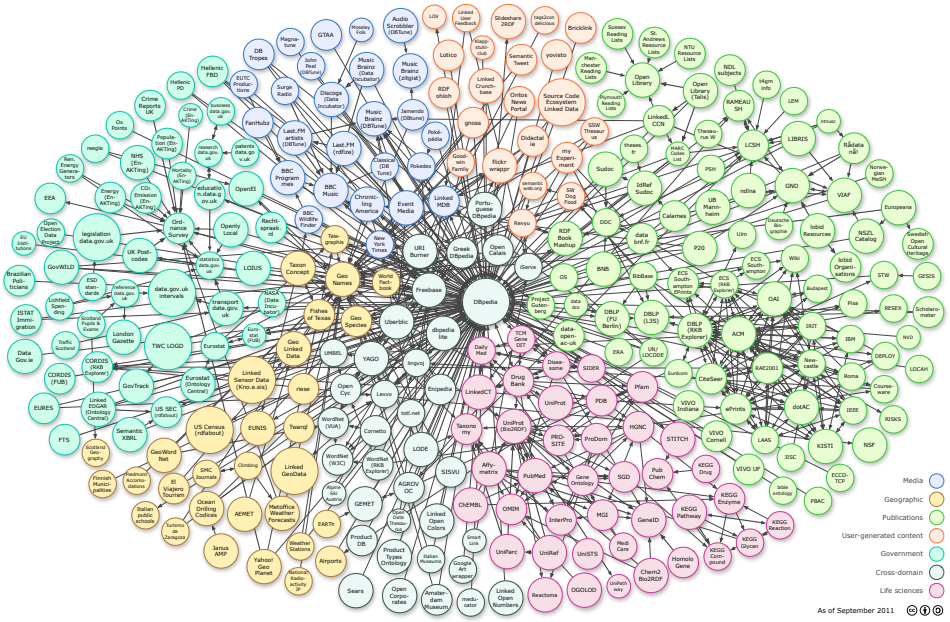


Figure 2.5: Linked Open Data cloud [18]

Lite, OWL DL and OWL Full. In each version, the expressiveness is addressed by posing a certain level of constraints and restrictions on the definition of classes and properties. The difference in allowed expressions between the different versions affects the computational runtime for the reasoning tasks. Using OWL is not necessary in all applications. Some applications, like the one presented in this thesis, do not require the advanced forms of expressiveness offered by OWL. In these cases, using RDF-S is sufficient.

Recently, the W3C has released the second and modified version of OWL, namely the OWL2 [8]. OWL2 comprises three profiles; EL, QL and RL using different reasoning technologies to provide computation time guarantees. OWL2 EL guarantees performing the reasoning in polynomial time. OWL2 QL uses database technologies to perform the reasoning guaranteeing a LogSpace complexity. OWL2 RL operates directly on RDF triples giving polynomial time responses.

2.2.3 Linked Open Data (LoD)

Following the Semantic Web conventions, many linked datasets, also referred to as Knowledge Bases, capturing knowledge from various domains were built in collaborative efforts like DBPedia[5], Yago[59] and Freebase[11].

In an effort to govern and evaluate the linked datasets, Tim Berners-Lee introduced a “5 Stars” scale for LoD [10] rating datasets on a scale from 1 to 5 according to the level of their usability. The first star is awarded for datasets whose data is open for public use.

The second star is awarded when the open data is structured in some machine-readable manner. If the structure follows a format e.g. CSV, the dataset is awarded a third star. The fourth star is awarded for datasets which in addition to the previous uses Semantic Web standards e.g. RDF and SPARQL and finally the fifth star goes for datasets which have links to other knowledge sets in order to provide context.

The latest number of datasets following the Open Data Principles is estimated to be around 295 and a diagram of these datasets and their interconnections is presented in Figure 2.5.

2.2.4 Named Entity Recognition

With the vast amounts of web content and resources whether documents, audio clips or video files, accurately describing such content becomes an important task in order for web users to easily find it. However, the metadata extracted from web content is restricted to keywords or labels which have no semantics. Without semantics, keywords could be misleading due to the presence of multiple semantic entities which share the same label. To elaborate with an example, consider the label "Apple". In a sentence like "Apple has announced the release of the new iPhone 6", the label "Apple" refers to Apple Inc. the American electronics corporation. On the other hand, in a sentence like "Fred has an Apple in his lunch box", the label "Apple" refers to the fruit. As the example suggests, defining web content through plain keywords only is not enough.

Named Entity Recognition (NER) or also called Entity Extraction is the process of mapping keywords or labels to the semantic entities they refer to from the context in which these keywords lie. From a Semantic Web and LoD perspective, NER links keywords in a sentence to KB classes and instances which accordingly provides more information about this keyword from KB knowledge. This process is of great importance because it provides means to enrich unstructured natural language web content with structured semantic keywords allowing better understanding and easier access [45].

NER tools use mainly Machine Learning techniques for detecting contexts and accordingly finding the respective semantic entities. Examples for such tools are the KIM tool introduced by Popov et. al in 2003 [51] and the NERD tool introduced by Rizzo and Troncy in 2012 [55].

2.3 Bayesian Networks

Bayesian Networks (also referred to as Belief or Causal Networks) are probabilistic graphical models which are most commonly used for uncertainty modeling. Their initial appearance was in the late 1970s motivated by the need to form coherent interpretation from the semantic expectations and perceptual evidence [50]. It first emerged in the field of medical decision systems [19] however its use has later widely spread to include various

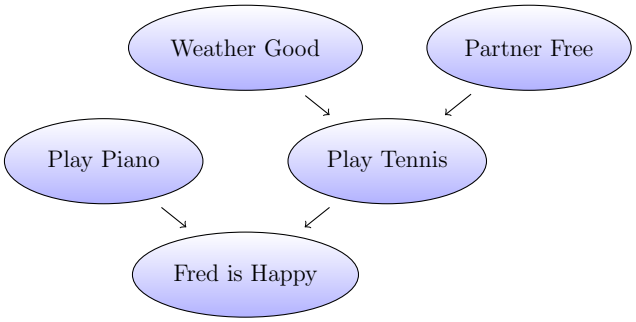


Figure 2.6: An illustration of a simple Bayesian Network

other fields.

2.3.1 Bayesian Network Structure

A Bayesian Network (BN) takes the form of a Directed Acyclic Graph (DAG) in which nodes represent random variables and directed edges represent the degree and direction of dependency/influence between the connected nodes. In that sense BNs provide both a qualitative representation of the domain problem through the structure of the network and a quantitative representation through the distributions describing the nodes and edges [21].

Figure 2.6 shows an example of a simple BN modeling the causal dependencies between Fred's happiness and the activities he did that day like Playing Tennis or Playing Piano. Knowing that playing Tennis requires some prerequisites which for simplicity in this illustration are limited to, whether the weather is good and whether Fred's partner is free to play. Such dependencies are visualized through the edges connecting for example the node "Fred plays Tennis" to the nodes which have a direct influence on it like "Weather is good" and "Partner is free".

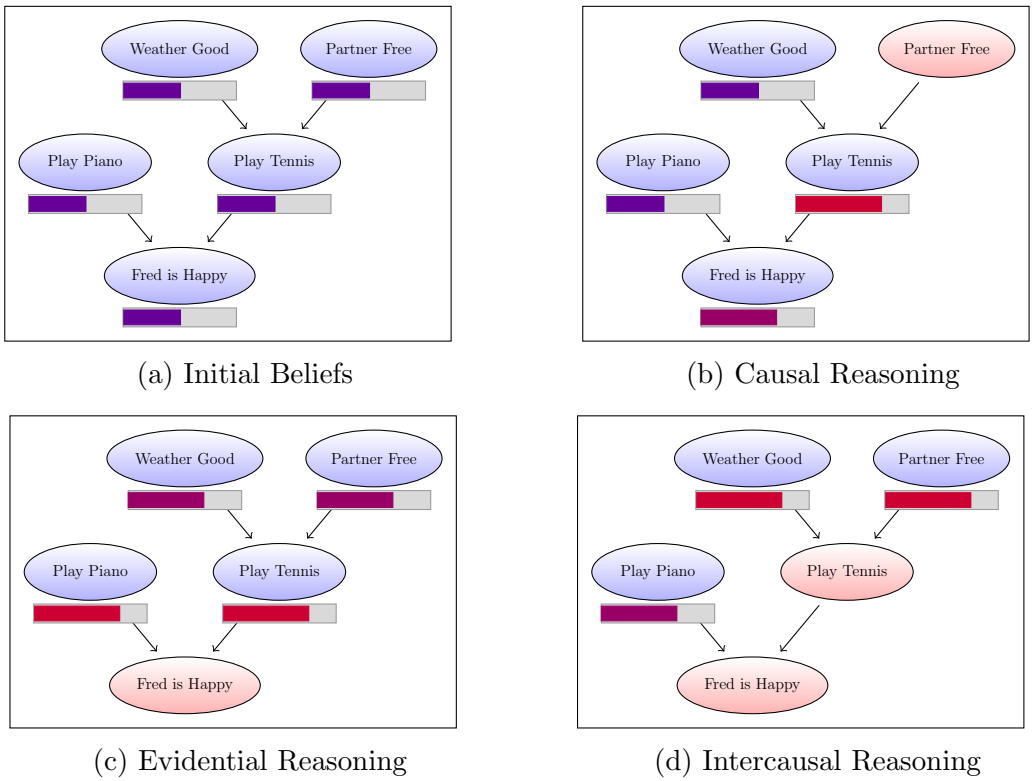
In a BN, the probability of each node X_i given its parents

$$\mathrm{P}(X_i \mid \mathrm{par}(X_i))$$

is expressed through a Conditional Probability Table (CPT) attached to this node, making the joint probability of the BN as a whole

$$\mathrm{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathrm{P}(X_i \mid \mathrm{par}(X_i))$$

The difference between BNs and other reasoning models is that in BNs, world knowledge and relationships are captured from real world, and represented in the model as abstract



(a) Initial Beliefs

(b) Causal Reasoning

(c) Evidential Reasoning

(d) Intercausal Reasoning

Figure 2.7: Illustrations for the different Bayesian inference patterns

and straightforward conditional dependencies as they are in reality, and the model is not affected at all by the reasoning process like in the other reasoning models [50]. It also considers only the known dependencies between variables making the reasoning process more efficient and flexible [17].

2.3.2 Reasoning Patterns in Bayesian Networks

Human reasoning is known to use the most complex reasoning patterns to make inferences and predictions out of the evidences available in real world. BNs support a set of various reasoning patterns resembling the human reasoning patterns. This section discusses some of these patterns.

Causal Reasoning

Causal reasoning also known as the prediction reasoning follows a top-down propagation of evidence [36]. That is, if the parent node has been observed, then the probability of the children of this node increases accordingly. For example in Figure 2.7b, having observed that Fred's partner is free (now marked in red) means that there is a higher chance that Fred plays Tennis which consequently means that there is a higher chance that Fred is happy. Or on the other hand, having observed that Fred's partner is not free means that there is a lower chance that Fred plays Tennis and so on. In this case, the observation of

the parents propagates to affect the probabilities of the children which in turn propagate these changes to further children (if any).

Evidential Reasoning

Evidential Reasoning or the so called explanation reasoning follows, in contrast to the Causal Reasoning, a bottom-up propagation of evidence [36]. This implies that, when a node is observed, the probabilities of the parents of this node increase in some sort of an explanation why the children have been observed. In other words, the parents are inferred. Figure 2.7c shows an example for Evidential Reasoning. In this case, it is observed that Fred is happy. As an inference to this observation, the probability that he played Tennis or Piano increases. Consequently, also the probabilities that the weather is good and that the partner is free increases.

Intercausal Reasoning

Intercausal reasoning is a reasoning pattern where independent nodes sharing the same children become dependent [36]. One form of the intercausal reasoning is when having some strong evidence about one parent causes the probability of other unobserved parents to decrease. This reasoning pattern is a very common form of human reasoning and is called *Explaining Away* [36] and this form is usually complex to model using the rule based models [50]. For example in Figure 2.7d, observing that Fred is happy and that Fred plays Tennis together causes the probability of Fred playing Piano to decrease since the explanation to Fred is happy has already been observed to be Fred playing Tennis. This hypothesis does not refute the fact that Fred may indeed have played both Tennis and Piano. However, it just assumes that a valid explanation for the observation has already been found thus reducing the probability of other possible explanations.

2.3.3 Inference Algorithms

Cooper described in his paper about the computational complexity of inference in BNs [17] two classes of BNs; singly and multiply connected BNs. Singly connected BNs are networks where each pair of nodes has a maximum of one path between them, whereas in multiply connected BNs, more than one path can exist between a pair of nodes. Usually for complex domains, multiply connected BNs are used.

Inference in BNs means computing the probability of the variable denoted by a certain node having one of its possible values given some observations about other related variables. Following are the inferences examined by the patterns in Figures 2.7b, 2.7c and 2.7d respectively.

$$\mathrm{P}(\text{Fred plays Tennis} = T \mid \text{Partner is free} = T) \tag{2.3}$$

$$\mathrm{P}(\text{Fred plays Tennis} = T \mid \text{Fred is happy} = T) \tag{2.4}$$

$$\mathrm{P}(\text{Fred plays Piano} = T \mid \text{Fred plays Tennis} = T, \text{Fred is happy} = T) \tag{2.5}$$

In [17], Cooper stated that the inference computation problem is solvable for singly connected BNs. Some algorithms exist which solve the inference for singly BNs in polynomial time [19]. Cooper proved however, that computing such inferences for the whole BNs in case of a multiply connected BN is NP-Hard. Following is a brief compilation of some of the most common Bayesian Network inference algorithms and approximations.

Message Passing Algorithm (1986)

Inspired by the human reasoning paradigms characterized by short term memory and narrow focus, Pearl [49] realized that humans tend to reason about things in a local manner first then gradually progress with reasoning along given relations. Therefore, he suggested a message passing algorithm in which he considers nodes to act as processors which hold belief values and communicate with neighboring nodes, and edges to act as pathways that control the flow of belief data. The inference problem is then approximated when each node passes the changes upon the newly given evidences to its neighbors until the whole network reaches an equilibrium.

One of the major advantages for such message passing paradigms is that they give a chance for tracking the flow of evidential data through the network and allows revising the intermediate steps.

Lauritzen and Spiegelhalter Algorithm (1988)

The Lauritzen and Spiegelhalter algorithm [39, 37] utilizes the fact that the inference problem is solvable for the singly connected BNs which can also be visualized as trees. The general idea is to convert the multiply connected BN into a tree of *local* clusters and handling the inferences first generally on the level of the clusters, which should be now easy given the tree structure, then locally within the clusters.

The clustering step is carried out first by connecting parents sharing the same child and dropping the edge directions. This step ensures placing the nodes with their parents within the same clusters. The resulting graph is then triangulated, that is, forcing the cycles in the graph to be of maximum 3 nodes by adding edges for cycles with more nodes. From the triangulated graph, the ranks of nodes are computed through a maximum cardinality search and clusters are formed such that each cluster forms a clique. The ranking of the nodes is used to manage the hierarchy of the tree of cliques. After the BN has been successfully transformed, the evidence is first propagated over the clique tree and then the single node belief is computed within the clique distribution. In [37], Kozlof and Singh suggest methods to parallelize this process in multiple points which makes it even faster and more efficient.

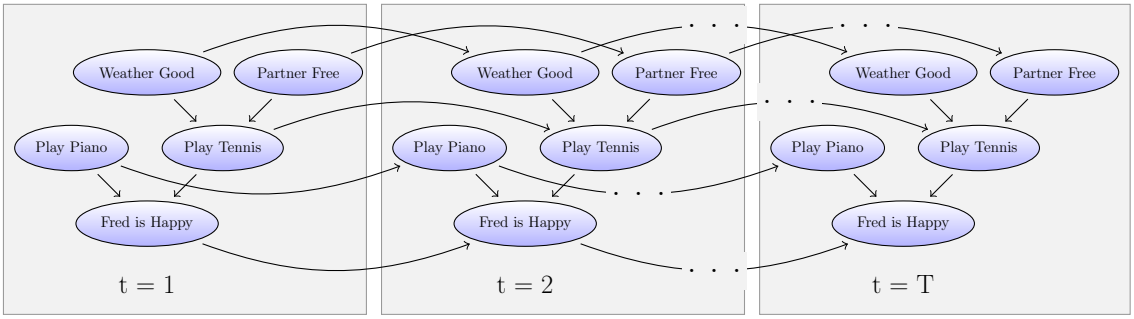


Figure 2.8: Example of an unfolded Dynamic Bayesian Network

Stochastic Algorithms (1988, 1990)

Another family of inference algorithms is the family of Stochastic algorithms in which the probabilities of observing certain values for variables are estimated from the frequency of observing them in simulation trials. Stochastic algorithms use approximations in contrast to the previously presented algorithms.

The baseline idea for this approach, logical sampling, has been introduced by Max Henrion in 1988 [32]. First, some values for the root nodes are randomly sampled based on the probability of their occurrence. Then recursively for each of the children, some value is randomly sampled weighted by the probability of this value given the parents. The sampling process is repeated till convergence. This approach however faces some problems when evidences are present since the samples that do not match the evidence are considered invalid and they get discarded. This increases the convergence time especially with increasing number of evidences. In the algorithm by Fung and Chang [27], they propose an extended stochastic probabilistic logical sampling approach which uses evidence weighting to overcome the evidences problem of the Henrion's logical sampling algorithm.

2.3.4 Dynamic Bayesian Networks

In a BN, it is assumed that given the same evidence and structure, the probabilities of variables represented by nodes is constant over time. This assumption is however not always true. Temporal reasoning has been introduced in 1988 by T. Dean [23] where he addressed the incapability of BNs of reasoning change over time and developed the so called Dynamic Bayesian Network (DBN).

A DBN is simply a BN, whose node probabilities at time t differ than that at time $t+1$. An exemplary structure of what an unfolded DBN may look like is the one shown in Figure 2.8. The major structural difference between BNs and DBNs is that each node in the DBN at time t has an extra dependency on itself or another node at time $t-1$ in contrast to the BNs whose dependencies all lie within the same time slice. DBNs are

used in applications which require dynamic reasoning over time under uncertainty, for example, Robotics.

2.4 Related Work

2.4.1 State of the Art Recommender Systems

Internet users use RSs while browsing and surfing the Web almost everyday. One of the leading RSs is the YouTube RS [20]. In the YouTube environment, the objective is to provide a diverse set of recent videos that are relevant based on the latest user activity. The YouTube recommender is considered a collaborative filtering RS. It uses the history of the user watching sessions to count for each pair of videos the number of times they were both watched in the same session. This way, for each video, the videos with high co-visitation count are considered related videos. Using this information gathered from other users, the set of recommended videos for a certain user is a subset of the union of the sets of videos related to the ones in the user's recent watch history. The ranking of the videos is generated based on video popularity and quality in a manner that ensures diversity.

Another very high traffic RS is the Amazon products RS [41]. Similar to the YouTube system, Amazon uses a collaborative filtering RS which performs an item-to-item similar tables based on the number of times both items were purchased together or within a short period of time. The product recommendations hence are just the aggregated set of items similar to the ones most recently purchased sorted by the degree of correlation or similarity.

Using Semantics

In the recent years, researchers in the RSs field decided to make use of the emerging technology of Semantic Web especially in content-based RSs where describing the content of items and building a model out of them plays an important role in contrast to the collaborative RSs where the content of the items plays almost no role. Using semantics instead of plain keywords for finding commonalities between items increases the chances of better *understanding* the user interests and consequently finding "similar" items to the ones which received a positive feedback.

Di Noia et. al addressed this in their LoD supported RS[25] where they transform content items into vectors of properties. The similarity between a pair of items is computed as the cosine of the angle between the two vectors and the relevance of an item is computed through heuristic functions in order to generate the final recommendations list. The results of their research showed good precision and recall values.

In 2009, Schopman et. al [56] introduced NoTube, a new system whose idea is to person-

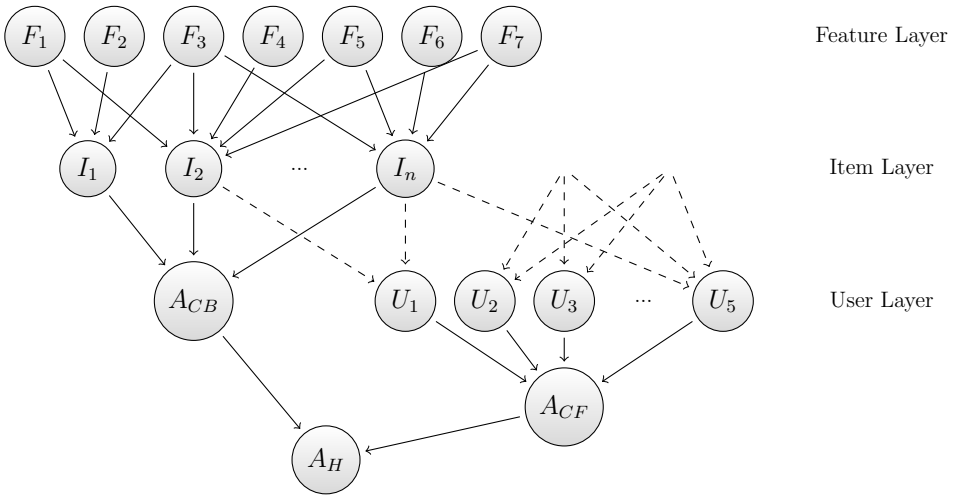


Figure 2.9: A Bayesian Network model for a hybrid RS [22]

alize TV experience through web services which use Semantic Web technologies to find relations between user interests captured from social networks like Facebook and Twitter, and TV programs. The authors however focused more on the conceptual aspect of the system and did not discuss much about the technical approaches they use for realizing these concepts.

Following the general concept of NoTube, the LinkedTV project [1] was launched in 2012. LinkedTV uses a content-based RS based on semantics to realize the personalization goal. Unique about the LinkedTV RS is that it uses its own Ontology; LinkedTV User Modeling Ontology (LUMO) to capture the content in the videos and uses it to model the user interests. Recommendations are generated first by filtering the videos which contain any of the entities in the user model or something semantically related to it. The filtered videos are then ranked using a heuristic function to compute the relevance of each video from its contents. The LinkedTV RS will be discussed extensively later in Section 3.1.

Using Probabilistic Models

Artificial Intelligence researchers have also realized the significance of using probabilistic models in problems that include high levels of uncertainty, like the problem of recommending content by guessing one's interests. In 2010, De Campos et. al [22] introduced a hybrid recommender system which uses a BN structure as the one shown in Figure 2.9 to infer the relevance value for an item depending on both its features and the feedback that similar users gave about it.

Chen et. al [14] have also introduced a RS for a digital library that uses BNs to realize relations between different books and accordingly direct the readers towards similar books.

[1] www.linkedtv.eu

The authors however did not clearly explain whether their system uses a content-based or a hybrid approach nor how the Bayesian network is created and on what basis the relations are established. But in general, the results reported in this paper were very promising.

2.4.2 Motivation of a New System

In this thesis, the objective is to develop a personalization component which recommends TV material for users according to their interests with major focus on news and talks. In this case, most of the items to be recommended are new items which have recently joined the system's library.

Collaborative filtering approaches like the ones used in YouTube [20] and Amazon [41] would not perform well in this case since the collaborative filters have the limitation of not being able to generate recommendations for new items which have a shallow history. Using a hybrid recommender in this case would be as well useless since it partially uses a collaborative filter. For this reason, it was decided to use a pure content-based approach to solve this problem.

The BN approaches presented in [22, 14] lack any semantic consideration about the items and their features. This means that really understanding the interests of a user through these systems is not possible and only keyword based recommendation is supported. The systems presented in LinkedTV and in [25], use content-based approaches and additionally use semantics allowing for more complex modeling of users interests. However, these systems use only heuristic models to generate the relevance values of items. Given that the user modeling problem comprises a huge amount of uncertainty, the heuristic approach in this case would be inferior to other more advanced learning approaches like BNs.

TV RSs are generally different than movie, music and book RSs. When recommending a movie or a book, parameters like movie actor or book genre plays the biggest role in choosing the item whereas the actual content plays a minor role. On the other hand, in case of news and talks, the content tends to be of more significance to the user. Therefore, available movie, music and book RSs are generally out of scope for this problem.

Trying to overcome the limitations discussed above within the scope of the targeted problem, this thesis proposes a semantically aware content-based RS that uses BNs for understanding users' interests and generating recommendations for them accordingly. The most related work known is the work presented in [4]. The system introduced in that paper however is not a RS, it is rather an interest discovery system that tries to capture the users' interests in a set of predefined topics. The interests are captured by semantically analyzing their social media content and reasoning under uncertainty in a BN. The work presented in this dissertation on the other hand does not involve recommending any items based on what was learned and the learning only supports a static set of predefined topics.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Chapter 3

Approach

In Fraunhofer IAIS, a RS has been developed as a part of the LinkedTV project. However, limitations have been observed in this system. This thesis introduces an alternative system that overcomes the limitations of the latter.

In this chapter, Section 3.1 covers the current implementation of the LinkedTV RS and points out its limitations. Section 3.2 introduces theoretically the proposed system and illustrates how it manages to overcome the limitations of the former system.

3.1 The LinkedTV Recommender System

LinkedTV as previously mentioned is an EU project which aims at changing the future of smart TV experience for users by tailoring the TV to users' interests and needs. The work in LinkedTV is divided among a set of components that interact with each other to deliver the full system functionality. Figure 3.1 shows the architecture of the full LinkedTV system.

In general the work flow starts by analyzing the videos added to the digital library through image and speech recognition techniques and annotating the videos by weighted semantic keywords extracted through named entity recognition as explained earlier in Section 2.2.4. This way, each video is annotated by a set of weighted *Video Annotations* (VA) which reflect the content of this video in terms of *semantic entities* i.e. classes and instances not just verbal keywords. Based on the VAs, more information related to the contents of the videos are imported.

On the other side of the system, the behavior of the users towards the content is analyzed, and according to the degree of attention observed as well as the explicit feedback given from the user, the degree of the user's interest in the content of the video is inferred. This information is modeled in the *User Model* (UM) also as a set of weighted semantic entities.

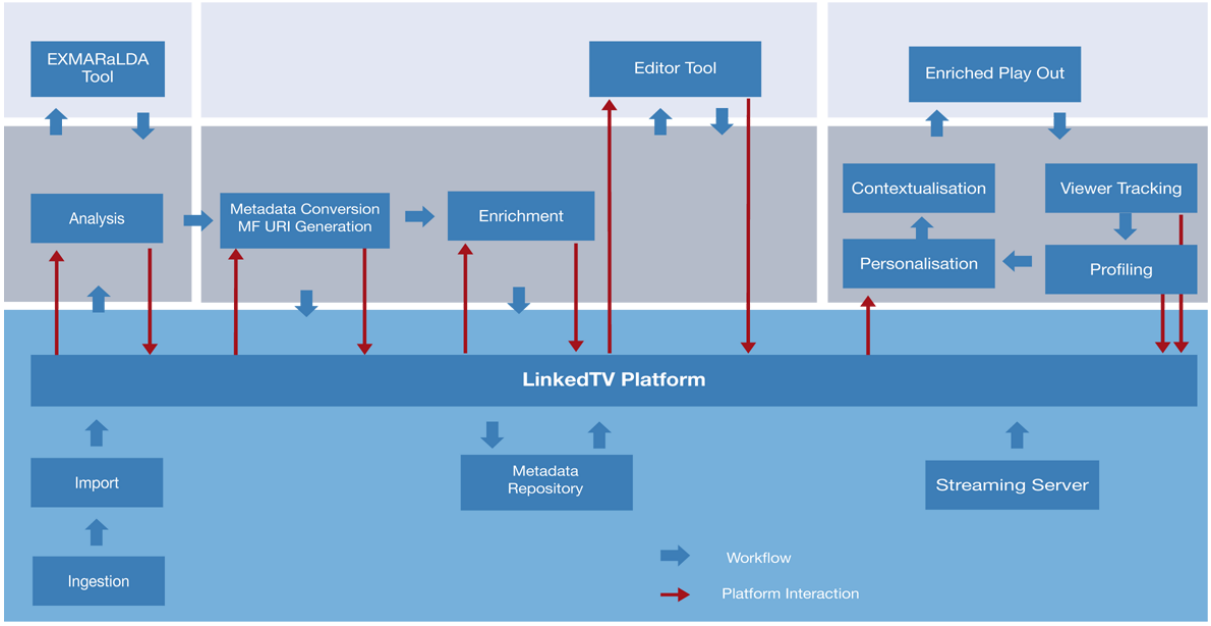


Figure 3.1: LinkedTV Architecture [15]

Given the information from both sides; the content analysis and the user behavior analysis, the personalization and contextualization components perform context-aware content-based filtering and recommends the videos that best match the user's interests.

Here in Fraunhofer IAIS, we were responsible for the development of the personalization and contextualization modules. The key feature of the personalization component was to filter out and recommend content as well as tailored side information about the content given a UM that defines the users' interests and dislikes. Therefore, it was decided to develop a content-based RS that uses semantic reasoning to filter and recommend content. Figure 3.2 shows the general work flow within the personalization component.

3.1.1 LinkedTV Knowledge Base

The knowledge in the world is massive, however it is hard, almost impossible, to find an ontology that models anything and everything in every possible domain. In a network of interrelated videos, in order to really understand the user's interests, the annotated content as well as the inferred interests have to be defined using a uniform and compact vocabulary. For this specific purpose, it was decided in LinkedTV to create a new domain oriented ontology [35].

The LinkedTV User Model Ontology (*LUMO*) is a compact lightweight OWL knowledge base (KB) that serves as the core for the services that LinkedTV provides. It encloses a coarse grained taxonomy of classes aligned with other LoD KBs. LUMO is designed in a manner that covers the general user interests and acts as the skeleton for the knowledge in

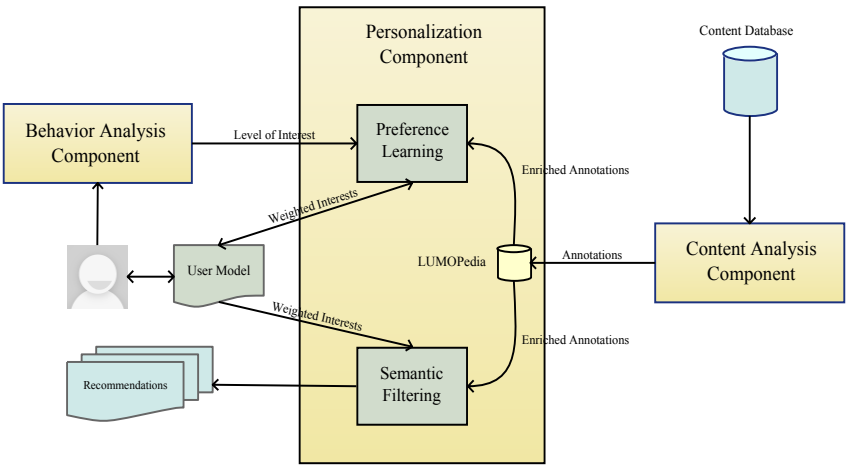


Figure 3.2: Workflow in the personalization component in LinkedTV

LinkedTV. The expressiveness of LUMO is restricted for simplicity to `typeof` relations to denote membership of an instance to a class, `subClassOf` relations to model the hierarchy of classes and a set of relations that are designed to connect the instances of these classes to each other. The weight of the relation symbolizes the significance of the relation to the subject and object entities. Examples for such relations for person instances are `bornAt`, `hasProfession`, or for location instances `isContainedIn`, `isBirthPlaceOf` or for topic instances `isSubTopicOf`, `isTreatedBy` ...etc.

LUMOPedia, the LinkedTV KB is based on the core ontology LUMO with additional instances imported incrementally and semi-automatically from LoD KBs like DBPedia [5] and Freebase [11]. The incremental import of instances allows controlled growth in the amount of information within the KB. The import is designed in a manner such that the instances expand with the new videos added to the digital library. That way, all the semantic annotations extracted from the videos are covered in LUMOPedia. On the other hand, the process is up to this point performed in a semi-automatic manner, the imported data is reviewed before being added to the KB. This ensures the quality of imported knowledge, since LoD KBs suffer sometimes from low quality information, due to the fact that they result from user driven efforts. The quality of user driven LoD KBs have been subject to the recent research project DBPedia Data Quality (DBPediaDQ) in 2013 [60].

Having defined the content of the videos in terms of LUMOPedia, the system can utilize the knowledge hierarchy to understand that if for example a user is interested in the general class Politician, a video about Angela Merkel or Barack Obama would be of interest to this user given the knowledge that Chancellors and Presidents are Politicians. Moreover the relations between instances allow the system to for example understand that if a user is interested in the English Premier Football League, a video about Chelsea and Manchester United would be of interest to this user because Chelsea and Manchester

United are teams which compete in the League. Worth to mention in this point that the expressiveness of the LUMOPedia ontology allows the users to formulate their interests in the form of semantic entities constrained by some relations for example Politicians bornIn Germany or American Presidents.

3.1.2 Semantic Filtering

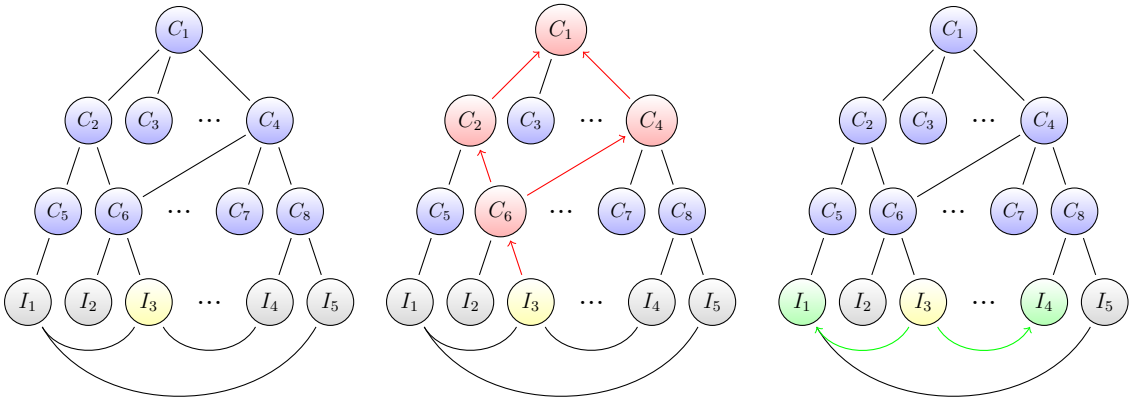
The LinkedTV RS, namely the *Personal Recommender*, is an in-database Postgres system where the LUMOPedia is stored in the form of relational database tables. The functions processing this data are also entirely coded in the form of database scripts. The personal recommender performs the semantic content filtering and accordingly generates it's recommendations to the user in two major steps. First, the list of VAs within a video is extended by including the entities that are semantically related to these VAs in the KB. Then, a relevance value is computed for each video by matching the result of the semantically enriched VAs with the UM. The following sections describe each step in more detail.

Semantic Enrichment

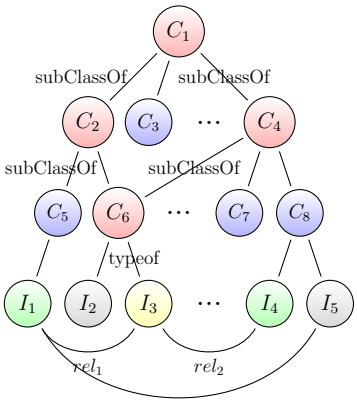
Since dealing with the complete LUMOPedia KB when matching the set of VAs to the UM is relatively complex due to the huge amount of facts in the KB, one way to simplify the matching process is to extract only the relevant pieces of semantic information needed for the matching by capturing a subset of the ontology that is related to the VAs and dropping the rest of the unused facts. In that manner, the matching process becomes more controllable.

Figure 3.3 shows an illustration of how the semantic enrichment of a VA entity is performed where classes are denoted by blue circles and instances are denoted by gray circles. In Figure 3.3a, the yellow node shows the start point that is the VA entity to be enriched. The first step of the enrichment shown in Figure 3.3b is to recursively extract all parent classes of this entity. The second step shown in Figure 3.3c is to extract all instances which have a relation connecting it to the VA entity within one hop in the graph.

The graph of extracted entities is used to form a list of weighted enriched video annotations (EVAs). The EVAs are the seed entity as well as the extracted entities in addition to constrained entities formed by making combinations of extracted parent classes with extracted related instances. Constrained entities are extracted to be matched with the ones formulated in the UM (if any). The weights of the enriched entities are determined heuristically as the product of the weight of the VA entity within the video and the weight of the relation connecting the enriched entity to the VA entity. Figure 3.3d shows an elaboration of this step.



(a) Original KB (b) Parent classes (c) Related instances



Entity	Weight	Possible Constraints
I_3	w_{va}	
I_1	$w_{va} \cdot w(rel_1)$	
I_4	$w_{va} \cdot w(rel_2)$	
C_1	$w_{va} \cdot w(\text{subClassOf})$	$C_1 \xrightarrow{rel_1} I_1$ $C_1 \xrightarrow{rel_2} I_4$
C_2	$w_{va} \cdot w(\text{subClassOf})$	$C_2 \xrightarrow{rel_1} I_1$ $C_2 \xrightarrow{rel_2} I_4$
C_4	$w_{va} \cdot w(\text{subClassOf})$	$C_4 \xrightarrow{rel_1} I_1$ $C_4 \xrightarrow{rel_2} I_4$
C_6	$w_{va} \cdot w(\text{typeof})$	$C_6 \xrightarrow{rel_1} I_1$ $C_6 \xrightarrow{rel_2} I_4$

(d) Result

Figure 3.3: Steps for semantic enrichment

Matching and Recommendation

Having simplified the problem in the previous step. The target of the Personal Recommender in this step is that given a set of weighted user interests in the UM

$$UM = \{(e_1, w(e_1)), (e_2, w(e_2)), ..., (e_n, w(e_n))\}$$

and a set of videos in a video database (VB) $\{V_1, V_2, ..., V_k\}$, each of which is annotated by enriched weighted entities

$$V_i = \{(eva_1, w(eva_1)), (eva_2, w(eva_2)), ..., (eva_m, w(eva_m))\}$$

where $w(x)$ is the weight of entity x, it is required to compute

$$\text{recommendations(UM)} = \arg\max_{i=1}^{k} \{\text{relevance}(V_i, \text{UM})\} \tag{3.1}$$

That is, the problem is narrowed down to computing the relevance function that matches the set of enriched video annotations in the VB to the set of user interests in the UM. A heuristic set of rules have been defined to compute the relevance function and are summarized as follows:

$$\text{relevance}(V, \text{UM}) = \sum_{i=1}^{m} \underbrace{\max_{j=1}^{n} \{eq(e_i, eva_j) \cdot w(e_i) \cdot w(eva_j)\}}_{\text{relevance of a single EVA to the UM}} \tag{3.2}$$

$$eq(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & otherwise \end{cases} \tag{3.3}$$

In that manner, the videos annotated with the largest number of high weighted entities related to the UM will stand out in comparison to other videos.

One of the very desirable features of the Personal Recommender is that it is a context-aware RS. In the current version of the system, the user manually inputs the contexts in which a certain interest is valid. In order to perform the recommendation within a certain context, only the subset of the UM interests which is valid in this context is considered for the matching process.

3.1.3 Limitations in Personal Recommender

Despite the good results obtained from the Personal Recommender of the LinkedTV system, it suffers a number of conceptual limitations which would affect it's performance for the general application with real data.

The first limitation lies within the semantic enrichment step. In this step, a subset of related entities is extracted from the ontology according to a set of rules. The extracted entities are only the ones related directly to the entity to be enriched and all of its indirect parents. However, this strategy imposes two challenges. First, the entities sharing a parent with the entity to be enriched for example the instance I_2 in Figure 3.3, or entities more than one relation hop away for example the instance I_5 are dropped from the enrichment. The assumption that such entities are not relevant may not always be accurate and may cause in some cases loss of information. Second, in an ontology, the level of relevance and detail within the hierarchy and relations is not uniform. This means that for some instances, the parents within 3 steps could be either still too detailed or already too broad as well as in relations where the instances within 3 steps could be totally irrelevant or very highly correlated. Therefore, controlling these aspects using

static strategies limits the use of semantics in this system.

Another limitation lies within the matching step where the relevance is computed from a set of formulas. In this step, the formulas shown in section 3.1.2 were chosen among a set of potential formulas according to their performance in a set of test scenarios. That means that the choice of the formulas was tailored on a specific narrow scope. This is considered a limitation because these formulas could be overfitting the problem, i.e. too specific for the matching problem and will not necessarily hold for other scenarios.

Observing the limitations above, a new approach was designed to overcome these limitations. More about this approach is presented in the following section.

3.2 The New Approach

Observing the nature of the addressed problem, it was found that the given preference learning and semantic filtering and recommendation problems are very analogous to pattern recognition problems. That is, given a set of training data which is the set of annotated content watched, as well as a set of semantic entities which represent how interesting the user found the content. The target is to learn the user's preferences i.e. learn the pattern by generating some user model such that when a new annotated content is given, the model can estimate how likely it is that the user will be interested in this content. It was also found that, considering the complete semantics tree of the video annotations in an unlimited manner, whether in terms of relations between or hierarchies of semantic entities, imposes an extra level of uncertainty to the problem addressed in this system. Consequently, in this new system, it was decided to exploit the recommendation problem using a learning approach to handle the uncertainty embedding semantics into the learning model in order to avoid the limitations mentioned earlier. In this system, the following assumptions are made

1. There is a VB in which each video is annotated by a set weighted concepts which capture the contents of the video.

2. There is a KB that correctly describes the hierarchies and relations between the annotation concepts.

3. The user interests are only content-related and are not affected by any other factors.

3.2.1 Knowledge Modeling

As explained earlier in Section 2.2, the semantic data within ontologies can be represented as a graphical structure in which nodes are entities and edges are relations connecting these entities together. With respect to LoD KBs, the nature of the inter-entity relations involve some level of uncertainty due to the questionable quality of data. On the other

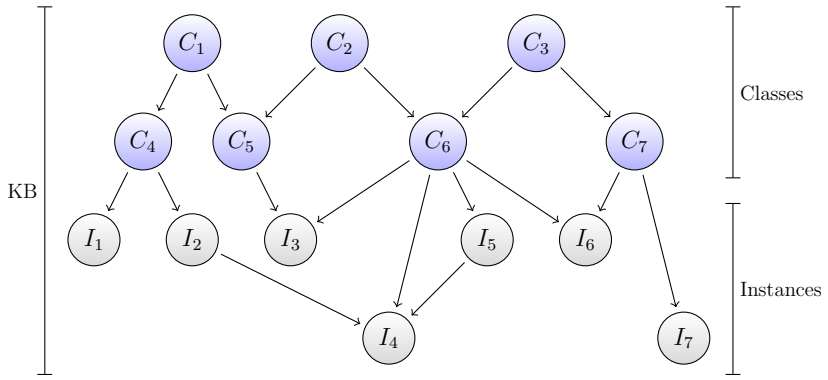


Figure 3.4: Example for modeling the user interests in terms of a BN

hand, in Section 2.3, it was shown that BNs have strong capabilities regarding the probabilistic graphical modeling especially under significant uncertainty. For the previous reasons, BNs were found a suitable candidate for the task of semantically modeling the user interests.

In general, humans tend to reason about interests in a structural topic oriented form where there are general topics and more specific topics. The causal reasoning pattern would infer that someone would be interested in a specific topic if this person has interest for the general topic. The evidential reasoning pattern on the other hand would predict that someone is interested in a general topic having observed that this person has interest for some of the more specific topics.

Similarly, the UM within a RS can be expressed in the form of an ontology which is transformed into a BN whose nodes represent the semantic entities denoting the interests and the edges represent the relations between them as shown in Figure 3.4. In this BN, each node has two states either "like" or "dislike" where the degree of belief of the node represents the probability of "like" or simply the negation of "dislike", and the dependencies are directed from the more general entity to the more specific. The information about which entity is more specific than the other can easily be extracted from the property relating the two entities. For example, the objects of the `typeof`, `isSubTopicOf`, `isContainedIn` and `subClassOf` relations are more general than the subjects since the classes are more general than their subclasses or instances overall. On the other hand, in the relations like `contains` and `isBirthPlaceOf` the objects of the relation are more specific than the subjects.

Following the idea presented in [22], it can also be assumed that the probability that a user likes a certain video depends on the degree of interest that this user has for whatever the contents of this video are. Since the contents of all videos are represented in the form of annotated semantic instances, the video nodes can be attached as leaves in the BN graph where each node is connected as a child to every instance it is annotated with as

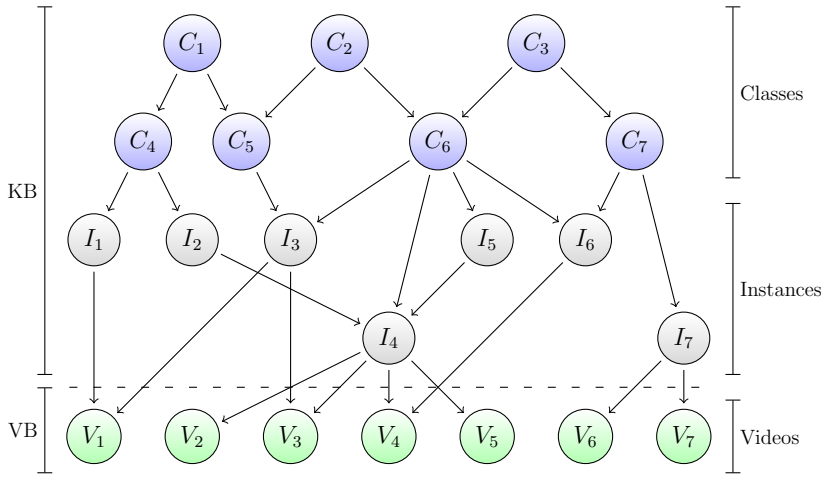


Figure 3.5: Example for adding VB nodes to the UM BN

shown in Figure 3.5.

Having a BN structure of nodes, a Conditional Probability Table (CPT) needs to be attached to each node as explained earlier in Section 2.3.1 to define the probability of this node given its parents. Following the illustration in [42], having a node n which has m parents, there are 2^{m+1} entries in the CPT of node n. Assuming that the m-dimension vector p represents the set of parents such that $p[i]$ denotes the status of the i-th parent, and that vector r represents the relation probability between n and its parents such that $r[i]$ represents the weight of the relation connecting n to the i-th parent. The entries of the CPT for all possible combinations of the vector p can be computed as follows where η is the system confidence factor.

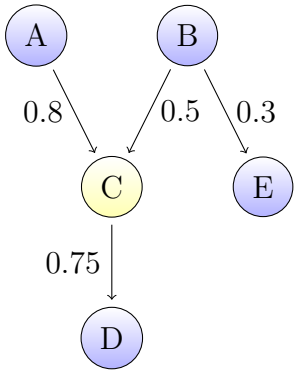
$$P(n = \text{Dislike} \mid p) = \eta \cdot \prod_{i=1}^{m} f(p[i]) \tag{3.4}$$

$$P(n = \text{Like} \mid p) = 1 - \eta \cdot \prod_{i=1}^{m} f(p[i]) \tag{3.5}$$

$$\text{where } f(p[i]) = \begin{cases} 1 & p[i] = \text{Dislike} \\ 1 - (r[i] \cdot \sigma(m)) & p[i] = \text{Like} \end{cases} \tag{3.6}$$

$$\text{and } \sigma(m) = 0.5 + e^{-m} \tag{3.7}$$

Notice in Equation 3.6 that the weight of the relation between the parent and the child is multiplied by a factor $\sigma(m)$. This factor is called the parent penalty factor and is introduced to sort of normalize the initial probability of nodes regardless of the number of parents. In order to better elaborate how the CPT for a node is constructed, Figure 3.6a shows an exemplary sample node in a BN (highlighted in yellow) whose CPT construction



(a) BN Snippet

Let $r_A = 0.8 \cdot \sigma(2)$, $r_B = 0.5 \cdot \sigma(2)$. Assume that $\eta = 0.9$

A	B	P(C=Like $\mid p$)	P(C=Dislike $\mid p$)
Like	Like	$1-\eta(1-r_A)(1-r_B)$ ≈ 0.7	$\eta(1-r_A)(1-r_B)$ ≈ 0.3
Like	Dislike	$1-\eta(1-r_A)$ ≈ 0.56	$\eta(1-r_A)$ ≈ 0.44
Dislike	Like	$1-\eta(1-r_B)$ ≈ 0.39	$\eta(1-r_B)$ ≈ 0.61
Dislike	Dislike	$1-\eta = 0.1$	$\eta = 0.9$

(b) CPT Construction

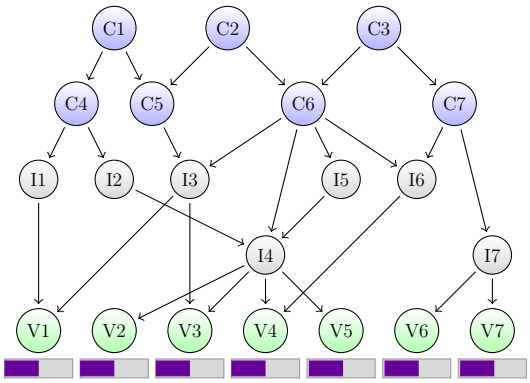
Figure 3.6: An example for CPT construction

is represented in the table in Figure 3.6b. Notice how the influence of the parent with stronger relation is higher than that of the parent with a weaker relation.

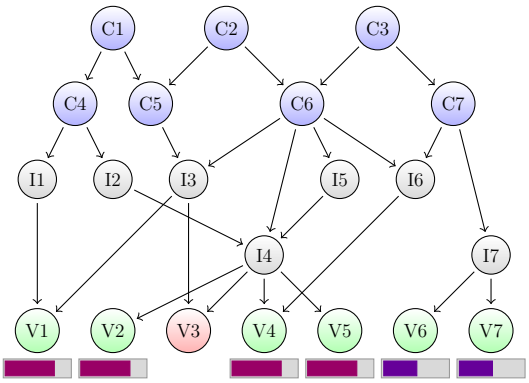
3.2.2 Learning and Recommendation

The following step after building a KB-based BN is to actually use it for learning and recommendation tasks. Recalling the nature of the learning problem, the given input is some implicit or explicit feedback from the user to one or more videos and the required output is to guess which are the topics or in other words semantic entities that the user has the strongest interest in. In the domain of BNs, this given is called *evidence*. Setting evidences in the BN, based on the user feedback, then recomputing the beliefs within the network will stimulate evidential reasoning starting from the leaves of the network which are the video nodes upwards towards the more general topics. Consequently with the change in beliefs for high level nodes, causal reasoning will trigger evidence propagation downwards towards other video nodes increasing the belief for these nodes relative to their distance from the originally observed one.

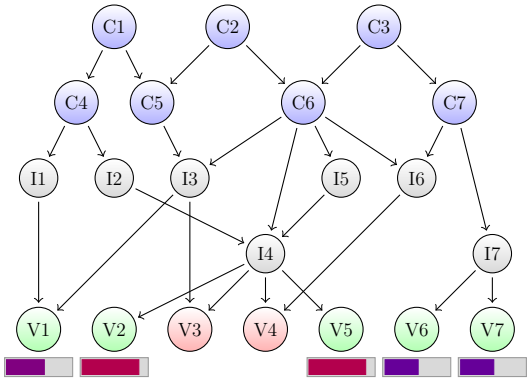
As demonstrated in Figure 3.7, the BN managed to detect the common semantic entities among the videos with positive feedback causing the belief in these entities to increase significantly. Consequently, the belief values of the video nodes connected to the same entities increase as well, as illustrated in Figure3.7b. Similarly, the belief values of the semantic entity nodes and the video nodes connected to them which are common among videos with negative feedback decrease significantly as shown in Figure 3.7d. This example suggests that with sufficient user feedback, not only can an approximate picture of the user interests be drawn through the evidence propagation within the network, but also the BN will be able to provide relevance values for each of the not yet observed videos.



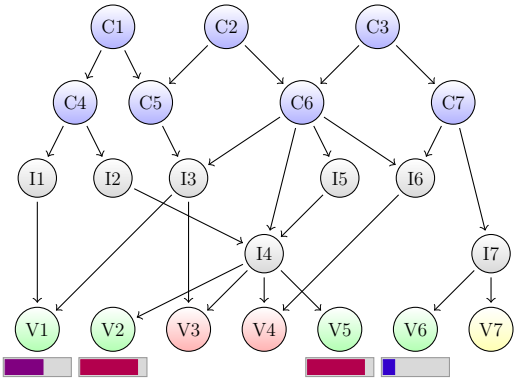
(a) Before evidence observation



(b) After observing positive feedback



(c) After observing another positive feedback



(d) After observing a negative feedback

Figure 3.7: Illustration of evidence propagation in KB-based BN

The example in Figure 3.7c also points out a very desirable behavior of the BN in this application which is the explaining away phenomenon explained earlier in Section 2.3.2. This behavior is desirable because it simulates how the real users think about video content. Usually, a user does not have to be interested in everything in a video to give it a positive feedback. The explaining away behavior in this case takes this perspective into account. Accordingly, once sufficient evidence is present for one of the parents of a video node, the BN reduces the belief for the other parents of this video node until more evidence is present for these parents.

In order to generate the recommendations for the user, simply the videos with the highest belief value are suggested either through a threshold filter or through a top-N technique.

3.2.3 Advantages and Limitations

Studying the behavior of the proposed approach, several advantages can be pointed out. First of all, it solves the limitations observed in the Personal Recommender system previously mentioned in 3.1.3. For example, the whole semantic tree is considered in this

approach without losing any information neither within the learning nor during the recommendation phases. At the same time, the matching of the user model and the video contents is not a complex matching task and it is no longer needed to develop a set of heuristic rules since matching in this approach is done implicitly through the propagation of the interest probabilities through the network. Furthermore, using the BN in this approach enables the system to perform complex reasoning patterns that are otherwise not possible through simple heuristics.

On the other hand, this approach suffers from some limitations. The first problem is the size of the KB. Since general purpose LoD ontologies are naturally very huge, it would not be feasible to create a user model based on a full LoD KB. Therefore, in order to be able to apply this approach, a concise domain oriented KB is needed to serve as the skeleton of the UM. Another problem could be faced in case of the presence of symmetric relations where both entities are within the same level of generality like for example in the `hasSpouse` relation. Moreover, the expressiveness of this model is limited to the level of the RDF-S. This means that complex OWL expressions as well as the relation constrained entries explained in Section 3.1.2 are not immediately possible in this model. Fortunately, some of these limitations can be ignored within the scope of the addressed problem or can be easily solved through application workarounds that will be explained in the following chapter.

Chapter 4

Implementation

In the previous chapter, the general architecture of the LinkedTV system was briefly outlined, the state of the art Personal Recommender of the LinkedTV system was presented emphasizing its limitations and a new conceptual approach that is capable of overcoming these limitations was suggested. In this chapter, an implementation of the proposed approach is presented, namely, the Bayesian Recommender System (BayRec).

BayRec is a three-tier web-application that uses a semantic BN-based RS in its logic layer and uses it's presentation to collect feedback from users. Generally, the low-level layers of the BayRec system is designed to fit within the general LinkedTV architecture presented in the previous chapter. However for the sake of completeness, BayRec is implemented as a standalone application to provide means to pass in input to the system and visualize the output which facilitates the debugging and testing processes. The architecture of the system is visualized in Figure 4.1.

The BayRec logic layer is developed as a separate **Java** module with a RESTful web-services API. For building the BNs and running the inference algorithms, the **jSMILE** library[1] is used. The library supports all inference algorithms mentioned earlier in Section 2.3.3. The presentation layer is simply an **HTML5** application built using the **AngularJS** framework[2]. The main goal of the front-end within BayRec is to compile the input needed and visualize the output produced by the back-end RS. Finally, the data layer uses **PostgreSQL**[3] to store the user history data, the semantically annotated video library and the domain oriented knowledge base. A set of procedural functions is specified within the database for fast and efficient information retrieval.

For the implementation purposes, the RBB dataset for news videos was used. This

[1] A Java wrapper to the SMILE library developed by the Decision Systems Laboratory of the University of Pittsburgh and available on https://dslpitt.org/genie/

[2] An open source structural framework for dynamic web applications available on https://angularjs.org/

[3] An open source relational database system available on http://www.postgresql.org/

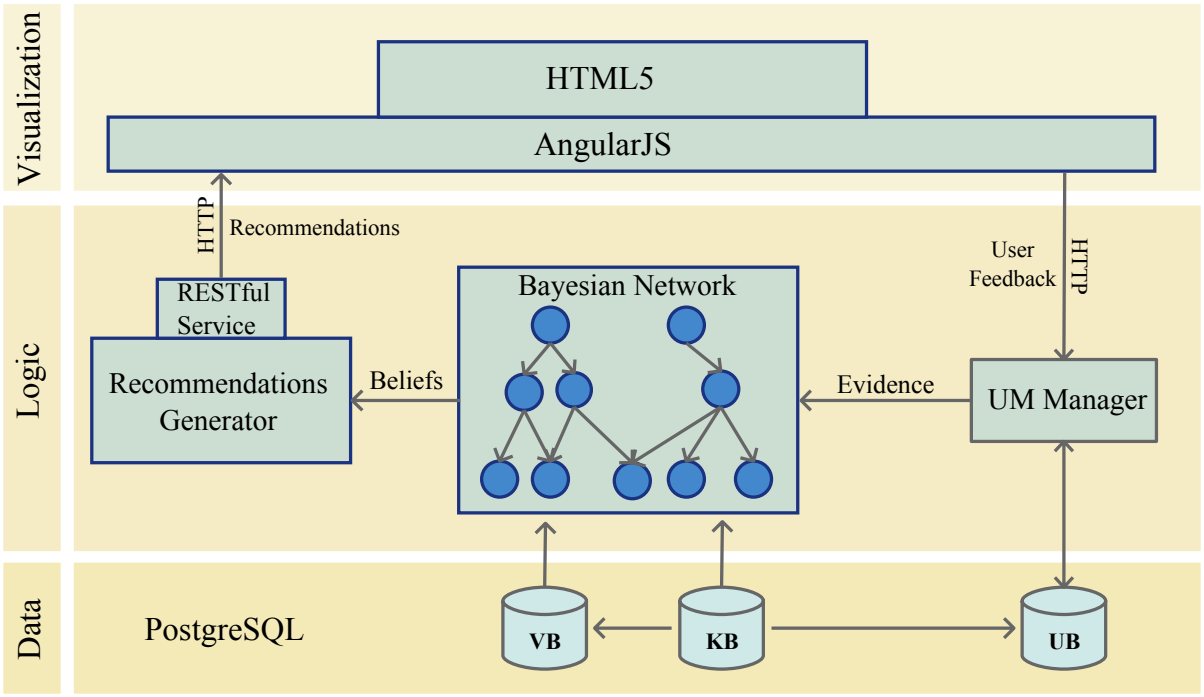


Figure 4.1: Architecture of the BayRec System

dataset was developed exclusively by RBB [4] for the LinkedTV project. It consists of 1024 videos whose annotations were automatically generated using the online TextRazor API [5].

The focus of this chapter is to shed light on the implementation details within the logic layer since it is the layer of interest within this system. First, the logic implementation of the baseline system is discussed in Section 4.1. Afterwards, some of the extensions applied to enhance the functionality of the system are presented in Section 4.2. The data layer of the system is not discussed in this chapter since it is out of scope for this thesis, but a briefing of the database schema is given in Appendix A. The front-end development details are also not covered in this thesis however a snapshot demo of the application is presented in Appendix B.

4.1 Baseline System Logic

Following the approach explained in the previous chapter, the functionality of the BayRec system is divided into three major steps as shown in Figure 4.2; first, building a BN structure which captures the relations between different semantic entities that may appear in the video contents from the LoD KB. The second step then is extending the BN with

[4] www.rbb-online.de
[5] https://www.textrazor.com/

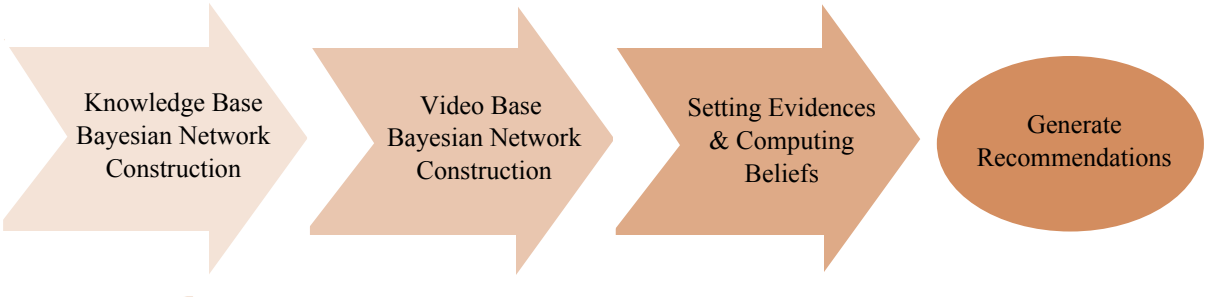


Figure 4.2: The steps of the recommendation generation process in the BayRec system

videos from the VB in a manner that reflects the general user's interest pattern. Finally, setting the evidences on this structure from the feedback collected from the user on some videos and updating the beliefs within the structure to generate recommendations. Observing that the KB rarely changes, an initial evidence-free BN structure can be generated offline once for all the users and regenerated whenever a change in the KB is required. Also noticing that the VB changes are not runtime changes, attaching the videos to the BN structure can be done offline once every 24 hours. The task of learning the user model and generating the recommendations would then be operated in runtime for each user on demand by setting evidences in the offline-generated BN structure and drawing conclusions out of the updated beliefs. The following sections demonstrates each step in more detail.

4.1.1 Bayesian Network Construction

As outlined earlier in Figure 4.1, all KB information is stored in a database supported by procedural functions to efficiently retrieve the data from the database tables. Therefore, the BN construction step is simply a transformation of the KB from relational database structure to a BN structure. This transformation is performed on two steps; first building a simple BN structure reflecting the semantic information from the KB. Secondly, adding and connecting the annotated videos in the VB to the outcome of the previous step. Algorithms 1 and 2 show the Pseudo code realizing these steps respectively.

In Algorithm 1, first, a node is created in the BN for each entity in the KB. Afterwards, for each of the created nodes, the parents are extracted from the KB and a corresponding arc is inserted in the BN. For classes, a parent can only be a parent class whereas for instances, a parent may be the class denoting the type of the instance or any other instance connected to the one examined with a parenthood relation. After extracting all parents, the CPT entries of each node are computed following Equations 3.4, 3.5 and 3.6. The result of this step is a structure similar to the one shown earlier in Figure 3.4 and is saved in a file to avoid the need for regenerating this structure when updating the VB. Similarly in Algorithm 2, all videos are fetched from the VB and for each video a node is

Algorithm 1 Building the BN-Structure from the KB

Create empty network BN
for all e where $e \in$ KB entities **do**
  Create node e in BN with 2 states "Like" and "Dislike"
end for
for all c where $c \in$ KB classes **do**
  for all p where (c `subClassOf` p) **do**
    Create arc (p, c) in BN
  end for
  Generate the CPT for node c in BN
end for
for all i where $i \in$ KB instances **do**
  for all p where (i `typeof` p) **or** (($p\ rel\ i$) **and** $rel \in$ parenthood relations) **do**
    Create arc (p, i) in BN
  end for
  Generate the CPT for node i in BN
end for
$BN \rightarrow$ `kb.xdsl`

Algorithm 2 Attaching Videos from the VB to the BN-Structure

$BN \leftarrow$ `kb.xdsl`
for all v where $v \in$ VB annotated videos **do**
  Create node v in BN
  for all a where $a \in annotations(v)$ **and** $a \in$ KB entities **do**
    Create arc (a, v) in BN
  end for
  Generate CPT for node v in BN
end for
$BN \rightarrow$ `vb.xdsl`

created and connected to the KB entities with which the video is annotated. The entries of the CPT for each video are again computed following Equations 3.4 through 3.6. The final outcome of these steps is a structure similar to the one shown in Figure 3.5 and is afterwards written and saved in a file to be used in the following step.

4.1.2 Learning and Recommendation

Initially in the BN-structure, all videos have approximately 50% probability of being liked for all users. Some videos have slightly more than 50% probability because they cover multiple topics therefore there is a higher chance that the user likes them. The first step in order to personalize the recommendations is to train the user model with the feedback given from the user on some of the videos. Algorithm 3 shows the steps for updating the user model. This process is performed in runtime for each user.

Algorithm 3 Learning the user's interest model

$H = \{(v_1, f_1), (v_2, f_2), ..., (v_n, f_n)\}$
$BN \leftarrow$ `vb.xdsl`
for all $(v, f) \in H$ **do**
  if $f = Like$ **then**
    Set $P_{BN}(v = Like) = 1$
  else
    Set $P_{BN}(v = Dislike) = 1$
  end if
end for
Update beliefs in BN

After running Algorithm 3, the beliefs within the BN will reflect the interests of the user in the form of the expected interest in each entity and video in the network. The second step then is to generate the video recommendations for the user. The algorithm proceeds in one of two forms (in BayRec specified by the user), either by suggesting the videos having the top N belief values as in Algorithm 4 or by suggesting the videos whose belief exceeds a certain value as in Algorithm 5. In a similar manner, the top N semantic entities and the ones with interest exceeding a threshold τ can be extracted from the BN.

4.2 Additional Features

Up to this point in the implementation, the system serves as a simple content-based video RS. However as explained earlier in Section 2.1.3, in real applications, users may require more complicated features for the system to be of real benefit to them. Some of the extensions mentioned are context aware recommendations and group recommendations. In this section, the implementation of both concepts within the proposed approach is presented.

Algorithm 4 Generating the top N recommendations

Require: BN beliefs are updated with user's feedback
  $R \leftarrow [\,]$
  for all $v \in$ video nodes in BN **and** v not watched by the user **do**
    if $R.size < N$ **then**
      $R \leftarrow R \cup v$
    else if $\exists\, v' \in R$ where $P_{BN}(v = Like) > P_{BN}(v' = Like)$ **then**
      $R \leftarrow R \cup v \setminus v'$
    end if
  end for
  return R

Algorithm 5 Generating the above τ recommendations

Require: BN beliefs are updated with user's feedback
  $R \leftarrow [\,]$
  for all $v \in$ video nodes in BN **and** v not watched by the user **do**
    if $P_{BN}(v = Like) > \tau$ **then**
      $R \leftarrow R \cup v$
    end if
  end for
  return R

4.2.1 Contextualization

In context-aware RSs, it is assumed that the interests of the user change according to the context in which this user is currently in. Usually, contexts are not mutually exclusive, that is, a user can have general interests which are valid in all contexts and some specific interests which are valid in certain contexts only. For example, a user can be generally interested in "Technology" but would rather be interested in "News" at home in the morning. In this case, the system should combine information about the general interests and context specific interests and recommend for example a video about "iPhone6 release mania". Contexts can be modeled in the form of a hierarchical structure to capture this type of relations since it offers different levels of specificity and consequently enabling the users to express their interests in an easier and more flexible manner. Allowing the users to choose between different contexts gives them control over which interests they would like to consider and which ones to ignore temporarily.

In order to add support for context aware recommendations following the proposed approach, the user model can be trained using only a subset of the videos which the user gave feedback about, this is the set of videos which were watched within the same context or a parent context, i.e. a more general one. In practice, this extension can be easily realized by adding extra information to the user feedback which denotes the context in which this feedback was given. While training the system, only feedbacks given within the

contexts related to the current context are used. This slight modification to Algorithm 3 is shown in Algorithm 6. In this version of the implementation, the contexts hierarchy is created manually by the user. Users are also expected to indicate in which context they currently are. However, techniques for automatic detection of contexts can be developed in future research.

Algorithm 6 Context Aware learning of the user's interest model

$cur \leftarrow$current context
$H = \{(v_1, f_1, c_1), (v_2, f_2, c_2), ..., (v_n, f_n, c_n)\}$
$BN \leftarrow$ `vb.xdsl`
for all $(v, f, c) \in H$ where $c = cur$ **or** $\exists parent(c, \text{cur})$ **do**
  if $f = Like$ **then**
    Set $P_{BN}(v = Like) = 1$
  else
    Set $P_{BN}(v = Dislike) = 1$
  end if
end for
Update beliefs in BN

4.2.2 Using External User Models

The presented implementation assumes that the input given to the system only includes the user's feedback on a set of videos. This is however not always the case. In some RSs, the users are able to specify their own UMs by explicitly setting certain topics or concepts to be interesting alongside the video feedback. The input is even sometimes only the UM without any video feedback like the case with the LinkedTV Personal Recommender.

Algorithm 7 Learning the user's interest model

$UM = \{(e_1, f_1), (e_2, f_2), ..., (e_n, f_n)\}$
$BN \leftarrow$ `vb.xdsl`
for all $(e, f) \in UM$ **do**
  if $f = Like$ **then**
    Set $P_{BN}(e = Like) = 1$
  else
    Set $P_{BN}(e = Dislike) = 1$
  end if
end for
Update beliefs in BN

The proposed approach can, with a simple modification to the learning algorithm, support using external UMs as shown in Algorithm 7. This can be achieved by setting evidences

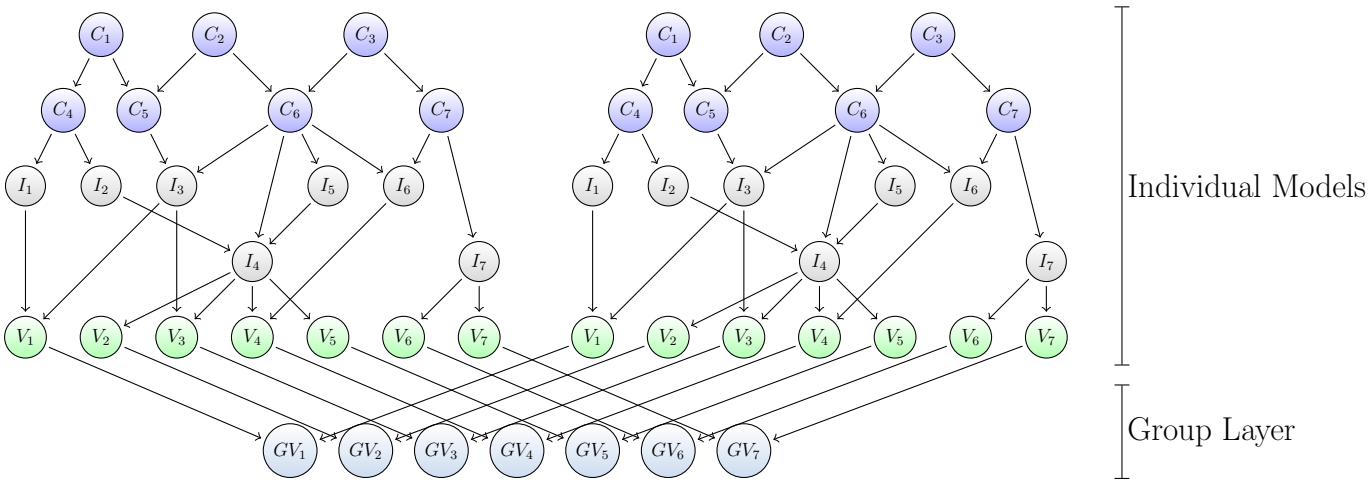


Figure 4.3: The structure of a BN for modeling group interests

in the higher level entity nodes according to the feedback of the user. However, in order to be able to do so, first the UM needs to be formulated in terms of a set of tuples, where each tuple represents an entity and the feedback to that entity. This modification, however, does not support the complex formed UM entries which involve relation constrained entities. On the other hand, this modification offers some sort of an integration point for merging both the proposed approach and the LinkedTV Personal Recommender.

4.2.3 Group Recommendations

Another feature of RSs that has been under research in the recent years, is the system ability to generate recommendations for a group, rather than for an individual user. In this case, a simple accumulation of interests for all users may be too naïve to solve this problem. Within the given approach, a simple extension can be added which provides support for generating recommendations for a group given the individual models of each group member as well as collective feedback from the group as a whole.

The extension relies on a slight addition to the user interests BN. First, the individual user interest BN for each group member is generated. Then, the different networks are connected together with an additional layer in the leaves of the network to represent the overall group interest in videos as shown in Figure 4.3. In that sense, the group interest in a certain video is depending on the interest of the individuals in this video which depends on their individual interests. In the same manner, the group liking a video implies that the individuals had some interest in the video which reflects on their individual user model BNs.

Chapter 5

Evaluation

In the previous chapters, the concept of a new content-based RS that uses semantics and BNs was proposed and the technicalities of an implementation of a RS which uses this approach were demonstrated. In this chapter, an evaluation of output obtained from the demonstrated implementation is presented. This chapter is divided into two sections. The first section discusses the evaluation of the output from the perspective of the quality of the recommendations. The second section focuses on evaluating the runtime performance of the RS within its different components.

5.1 Recommendation Quality Evaluation

Evaluating the quality of the recommendations given by a recommender system has been a research topic on its own in the past decade. Many papers such as [57, 33] discussed several metrics by which a RS can be evaluated. Section 5.1.1 offers a general theoretical briefing inspired from [57, 33] about the metrics that were chosen to evaluate the proposed RS explaining the reasons why each metric was found relevant to the problem. Afterwards, the setup of the testing experiments is explained in Section 5.1.2 and the results expressed in terms of the previously explained metrics are presented in Section 5.1.3.

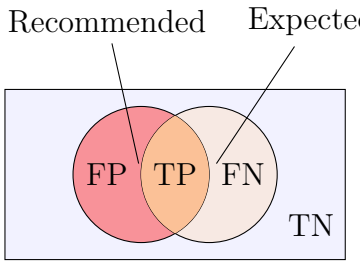
5.1.1 Quality Metrics

Given the Information Retrieval (IR) nature of the problem, the immediate metrics that come to mind are **Precision**, **Recall** and **Accuracy**. Given the factors shown in Table 5.1, the Precision, Recall and Accuracy values can be expressed as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{5.3}$$



	Recommended	Not Recommended
Interesting	TP	FN
Not Interesting	FP	TN

Figure 5.1: The factors which contribute to IR metrics

In general, Precision is used to compute how many items are relevant within the returned set of items whereas Recall represents how many items from the expected answers were returned. Accuracy normally judges the system overall by computing the percentage of correct decisions taken. In the context of the RS under test, the Precision plays a more important role than the Recall because normally the aim is to return a concise set of relevant videos rather than a huge set which covers all interesting and uninteresting videos to the user. However, the Accuracy can give an indication about how well the system is generally performing because it takes into consideration the size of the video database from which the recommendations are chosen.

Another factor that plays a great role in RSs is **Diversity**. In RSs, the Diversity metric measures how variant the recommendations generated by a RS are. It is possible to return a list of items all annotated with the exact same thing, Precision in this case would be very high. This, however, does not give the user any chance to explore something related but different, that is why the this metric is important in RSs. Diversity within a recommendations list of size N can be computed following [24] as shown below where $sim(x, y)$ is a function which expresses how similar two items are on a scale of $[0, 1]$ where 0 is unrelated and 1 is identical.

$$\text{Diversity} = \frac{1}{2}\sum_{i=0}^{N}\sum_{j=0}^{N} 1 - sim(i, j) \tag{5.4}$$

Some RSs have huge VBs however only a few of them can be used for recommendations due to some prerequisites which have to be present in a video in order for it to appear as a potential recommended item. Such prerequisites can be a minimum number of views to guarantee popularity or a minimum number of reviews in case of collaborative RSs. The **Catalogue Coverage** metric computes the percentage of potentially recommended items in the VB. Having a metric to evaluate this aspect is important because it gives an indication of the flexibility of the RS.

5.1.2 Test Setup

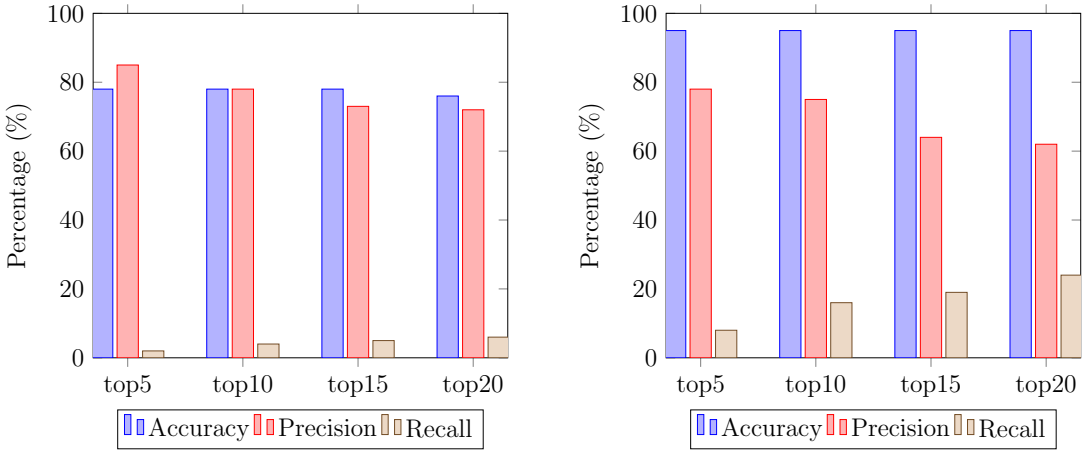
In order to evaluate the proposed RS against the previously mentioned metrics, the TED dataset from April 2012 [47] later referred to as TED-1 and the one from September 2012 [48] later referred to as TED-2 were used as VBs with some slight modifications. TED-1 consists of 1149 videos each annotated by a some tags out of 300 possible tags while TED-2 consists of 1203 videos each annotated by some tags out of 300 possible tags. The tags however were not weighted by their significance to the video therefore, assumptions about the weights of the tags were added randomly to cover this missing data. As a result, the user collected feedback was no longer usable since the assumed values for tag weights were not truly reflecting the content of the video on which the user gave the feedback.

For measuring the Precision, Recall and Accuracy, a KB with approximately 500 classes and 400 instances was setup and a set of videos annotated with one of the tags was used as observation. The reason for using a set rather than a single video was to avoid a cold start and to cause a cluster of interest. The expected set of recommendations was considered as the videos which are annotated with the same tag or one of its semantically related tags. In each test, the system would take as input the set of observation videos and return recommendations with different recommendation settings i.e. Top N and threshold and using different inference algorithms namely the Lauritzen and the Logical Sampling algorithms. The output of each run would then be compared to the expected set to compute the values for the precision, accuracy and recall.

5.1.3 Results

In the first test, 300 runs, one for each of the possible tags, was performed using the Lauritzen inference algorithm, generating the top N recommendations where N ranges from 5 to 20 with a step size of 5. This test was repeated for each dataset, and the precision, recall and accuracy values were computed for each of these runs. The average of the computed values per dataset are presented in Figure 5.2. The results of this test show that the system achieved an almost constant accuracy of around 78% in the TED-1 dataset and 95% in the TED-2 dataset. The precision values showed a decrease as expected with the increase in the number of recommended items since the set becomes less concise. However, considering the Top-5 and Top-10 methods in Figure 5.2a, they achieve an average precision of 85% and 78% respectively. This means that almost 8 out of every 10 videos recommended should “theoretically” be an interesting video.

In another test, the same procedure of the previous test was performed but using the threshold method with threshold values ranging between 70 and 90 with a step size of 5 for generating the results instead of the Top-N method. The average values were computed per dataset and are presented in Figure 5.3. Worth to mention at this point is that the averages for the higher threshold values like 85% and 90% are computed only in the cases when the recommendation list was not empty. This however was not a very



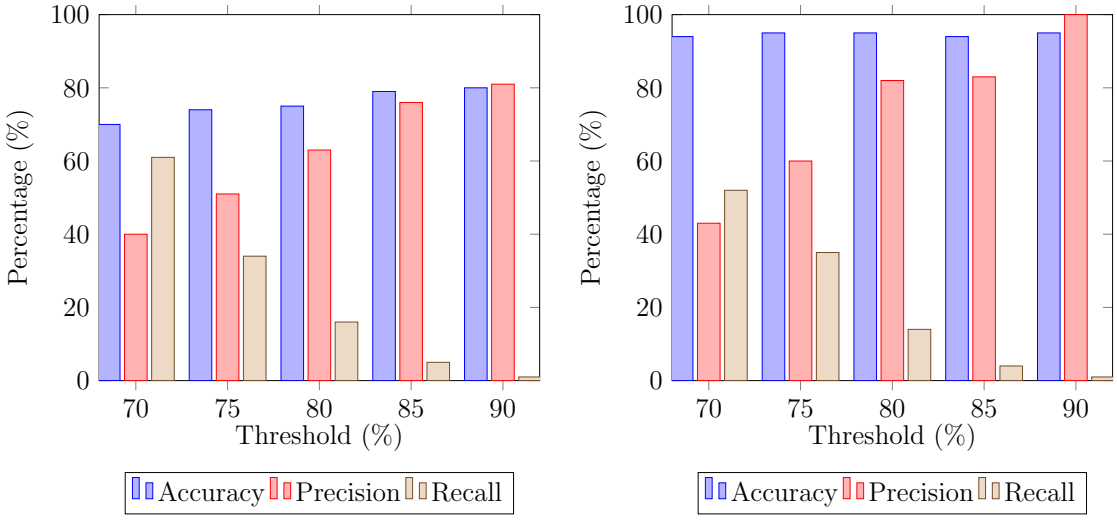
(a) TED-1 (b) TED-2

Figure 5.2: Average precision, recall and accuracy values for Top-N method

common case because it was very rare that the recommendation value for some videos exceeded these values, and when it did, then the precision value was 100% because it was definitely an interesting video from a theoretical point of view. But in general, the results of this test confirm the findings from the previous test that the larger the number of recommended items in this case controlled by lowering the threshold value, the less precise the set becomes but contrastingly the higher the recall value since more videos can make it to the recommended list.

In a third test, the target was to compare the quality of the recommendations generated by different Bayesian inference algorithms. Therefore, the same procedure of the previous two tests was repeated using the Logical Sampling algorithm instead of the Lauritzen algorithm used previously. The comparison of the results obtained from each algorithm is shown in Figure 5.4. It is clear from the figures that the Lauritzen algorithm performed better in both datasets than the Logical Sampling algorithm. A possible explanation for this finding is that as discussed earlier in Section 2.3.3, the Logical Sampling algorithm is a stochastic algorithm that approximates the belief values in contrast to the Lauritzen algorithm which actually performs exact inference by some structural transformations. Therefore, the inaccurate approximations could explain the inferior quality.

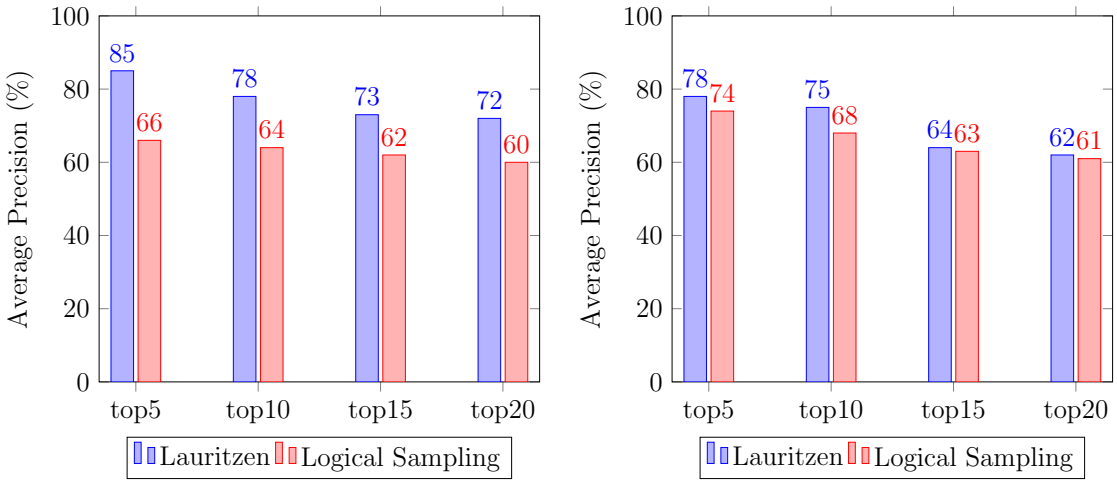
In addition, an extra test was performed to study the effect of the choice of the BN inference algorithm on the diversity of the recommendations. Therefore, for each of the runs of the previous tasks, the diversity of the recommended list was computed according to Equation 5.4 where the similarity function counts the number of overlapping annotations. Figure 5.5 shows the average diversity value achieved per method in each algorithm against the precision. As shown in the figure, the diversity of the recommendations list increases as the number of recommended items in the list increases which means that it is inversely proportional to the precision value. It is clear also from the figure that the



(a) Results obtained from testing on the TED-1 dataset

(b) Results obtained from testing on the TED-2 dataset

Figure 5.3: Average precision, recall and accuracy values for threshold method



(a) Average precision values obtained from tests on the TED-1 Dataset

(b) Average precision values obtained from tests on the TED-2 Dataset

Figure 5.4: Average precision values obtained using different inference algorithms

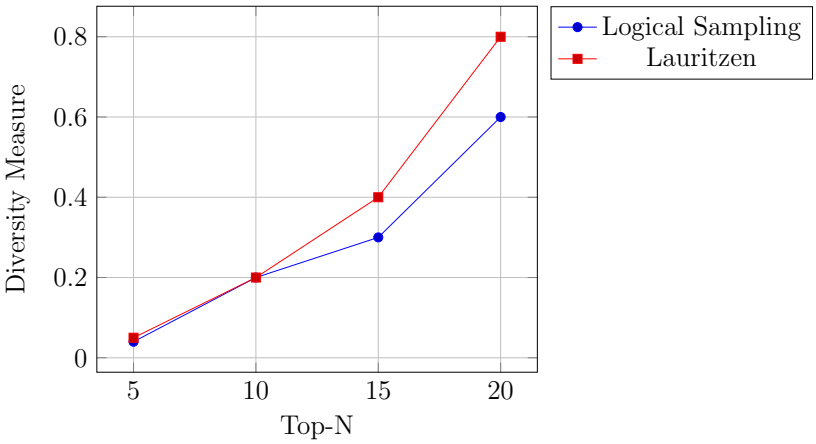


Figure 5.5: Effect of inference algorithm on recommendations diversity

recommendations generated based on the beliefs computed by the Lauritzen algorithm are more diverse than those by the Logical Sampling algorithm.

In general, the Catalogue Coverage of the proposed system is the number of videos in the VB which have at least one annotation which describes the content of the video. In case of the given datasets; TED-1 and TED-2, the Catalogue Coverage is 100% because there are no constraints on recommending any of the videos if some related video is observed.

5.2 Performance Evaluation

In any system, judging an approach by its quality only is not enough. The runtime performance is another critical side that is sometimes even more crucial than quality especially in the cases of online systems which require real time results. In this section, first a theoretical analysis of the runtime complexity for each process of the system is analysed in Section 5.2.1. In Section 5.2.2, the setup of the runtime performance experiments is elaborated and then Section 5.2.3 presents the results obtained from the runtime performance experiments.

5.2.1 Theoretical Analysis

The first step of the process that was previously explained in Algorithm 1 is the construction of the BN which represents the KB. In this step, the time complexity of maximum $O(n^2)$ in the worst case which is logically impossible that every node is having dependency on every other node, and $O(n)$ in the best case which is the trivial case that every node is not having any parents where n is the number of entities in the KB. On average, the time complexity of this step would be $O(np)$ where p is the number of parents per node which is usually small.

After constructing the BN from the KB, the videos in the VB are added to the BN following the Algorithm 2. The time complexity of this step is again O(vn) in the worst case scenario where each video is connected to all the nodes and O(v) in the best case when every video has no parents where v is the number of videos. Also this step on average has a complexity of O(vm) where m is the number of annotations per video and is normally not very huge.

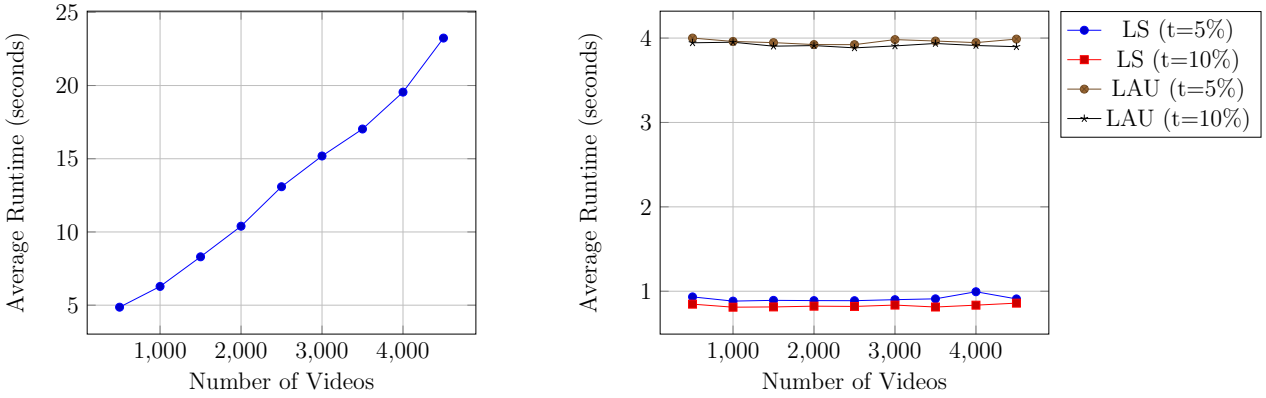
Afterwards, the major step in the process is the step of updating the beliefs in the BN after setting the evidences observed from the user's feedback using the BN inference. In this step, the runtime complexity depends on the BN inference algorithm used. In the case of using the Lauritzen algorithm, the runtime complexity is O(2^n) where n is the size of the largest clique in the transformed graph generated from the BN [31] whereas in the case of using the Logical Sampling algorithm, the time complexity is independent on the size of the BN due to its sampling nature as explained earlier in Section 2.3.3. However in this implementation, the inference is performed as a black-box within the SMILE library therefore these complexities are just theoretical.

Having updated the beliefs in the BN, the final step is to read the values of the updated beliefs and generate recommendations according to the chosen method whether threshold or Top-N method as elaborated earlier in Algorithms 4 and 5. In case of the threshold method, the runtime complexity is O(n) where n is the number of nodes in the network. However in case of the Top-N method, the complexity would be O(nN). Therefore, it only makes sense to use the Top-N method for small N otherwise, the threshold method would be more convenient.

5.2.2 Test Setup

In order to verify the previous complexity analysis, the runtime of each test was recorded for different sizes of the VB, different sizes of the KB and different recommendation settings. The VB was composed of the TED-1 [47], TED-2 [48] datasets in addition to the RBB dataset together. The KB consisted of 500 classes and 400 instances. The possible inference algorithms were the Lauritzen and the Logical Sampling algorithms. The runtime experiments were performed on an Intel(R) Core(TM) i5-3317U CPU @1.70GHz processor accompanied by 4GBs of RAM.

In order to control the number of videos in the VB, only a subset of the actual VB was considered when adding the videos to the BN while keeping the KB in its original state which ensured that all the parents of the video nodes would be available in the BN. On the other hand, it was not feasible to apply the same technique for controlling the size of the KB due to the complex relations between the nodes. Therefore for simplicity, the scalability tests for the KB were ruled out from this set of experiments.



(a) VB construction runtime (b) Recommendation generation runtime

Figure 5.6: VB scaling effect on the runtime performance

5.2.3 Experimental Results

The focus of the performance experiments was to analyze the runtime behavior with respect to the size of the VB while having the KB size constant and set to the maximum size available. Therefore, in the first experiment, the average time in seconds needed for constructing both the KB and the VB was computed from 10 runs for each of the VB sizes ranging from 500 videos up to 4500 videos with a step size of 500. On the other hand, the focus of the second experiment was to study the runtime behavior towards the growth in the size of the VB in the step where the beliefs in the BN are updated based on evidences under different Bayesian inference algorithms. The time consumed to generate the recommendations for observation sets representing 5% and 10% of the total size of the VB using both Lauritzen and Logical Sampling algorithms was recorded.

The results obtained from both experiments are shown in Figure 5.6 where Figure 5.6b shows the running times recorded in the construction phase of the BN with respect to the VB size and Figure 5.6a shows the average time in seconds needed to generate a recommendation list with respect to the VB size.

The results in Figure 5.6a show a linear growth of runtime with the linear increase in VB size. This result confirms the theoretical analysis of this process explained above. Given that the construction of the VB is performed offline, this can be considered as an acceptable runtime growth. Furthermore, the durations recorded in the second experiment show an almost constant runtime for the belief updates regardless of the size of the VB and the size of the evidences list t with respect to the VB size. Note however that the Logical Sampling (LS) in this case performs almost four times faster than the Lauritzen (LAU) algorithm. This behavior is explainable for the Logical Sampling algorithm as discussed earlier in Section 5.2.1. The behavior of the Lauritzen algorithm can be justified that the horizontal increase in the number of leaves in the BN does not affect the size of the largest clique therefore not affecting the algorithm runtime.

Chapter 6

Conclusion

With the horrendous increase in the amount of multimedia content in the web, a cutting edge solution was needed to enable end users to easily reach the multimedia content that match their interests. For this purpose, RSs were developed to direct the users through recommendations to what might be of interest to them. Focusing on video RSs, it was found that some of the state of the art RSs like for example YouTube rely heavily on finding similarities between users and accordingly recommending to them the videos that similar users found interesting. This approach, however, does not take into account the contents of the videos nor the actual interests of the users. Alternatively, RSs like the LinkedTV PR tried to overcome this limitation by recommending videos whose content match the user interests. Although the matching process was designed to use semantics rather than just keywords for matching, the process was performed in a rule based manner relying mainly on the interests explicitly specified by the user.

Examining the nature of the RS problem, it was immediately noticed that the problem is a Machine Learning problem, which deals with various sources of uncertainty, especially with respect to the semantic KB which defines the relations between the contents of the videos. Therefore and in an attempt to further exploit this problem and extend the available RSs aiming to overcome their limitations, the concept of a new semantically aware RS that uses BNs instead of heuristics was proposed, implemented and evaluated in this thesis.

Built on the assumption that if a user is interested in some entity, then this user would also be interested in a video that handles this entity, the proposal was to simply transform the problem into a huge BN where the nodes of the network represent the degree of interest of the user for some entity. The VB videos were used as leaf nodes where the parents of each video node are the entities which describe this video. Finally the problem was solved by setting evidences in the BN using the user feedback and performing reasoning on the BN using inference algorithms to compute the degree of interest of the user for the unobserved videos in the VB given the observed evidences.

Several tests have been carried out to verify the feasibility of the above mentioned ap-

proach from the quality and the performance points of view. The quality tests have shown an average accuracy of 75% up to 95% and an average precision between 60% and 80% depending on the Bayesian inference algorithm used. Considering the relatively high accuracy and precision values, it can be concluded that the proposed approach performs generally well for the addressed problem quality-wise. From a performance point of view, it was proven by practical experiments that the system scales very well with respect to the size of the VB with an overall runtime for the recommendation process of approximately 1 to 4 seconds for a compact KB also depending on the Bayesian inference algorithm used. From this result, it can also be concluded that the above mentioned approach is feasible for implementation in real life applications. Following the tradition, the inference algorithm which performed better in terms of quality consumed more time following the quality-performance trade-off.

Nevertheless, despite the good results that were reported, this approach has been observed to have several limitations. Section 6.1 highlights the limitations and then ideas about how to extend the work presented in this thesis are offered in Section 6.2.

6.1 Limitations

The conceptual limitations as discussed earlier in Section 3.2.3 can be summarized as follows; first no symmetric relations within the KB can be modeled using a BN because in such cases, it is not possible to identify whether the subject or the object is the parent or the child. Second, the presented model is limited to the RDF-S expressiveness however it is not immediately clear how to model more complex class definitions like the ones allowed in OWL or how to model constrained interests like the ones allowed in LinkedTV. Also, the proposed model does not take into consideration the temporal information neither about the observations nor about the videos themselves. This is a loss of precious information because usually the temporal sequence of feedback can lead the system to draw conclusions about the user's interests which cannot otherwise be captured, and the temporal information about the videos can sometimes be needed to form a bias towards the more recent videos as potentially more interesting than older ones. Another essential limitation is that the weights determining the strength of the dependency between two nodes is currently static.

Furthermore, some limitations were discovered during the evaluation phase of the system. One of which was that when a user has two major interests where the interest weight of one of them exceeds the other, the recommender system will first try to find videos which contain both interests together which is a desirable behavior. However, if such videos are not available, the Top-N recommendations will be dominated by videos fitting the interest with the higher weight causing the diversity of the recommended videos to be very low. Such behavior is not very desirable in RSs since one of the objectives of the RSs is to enable the user to explore new diverse content not to stay within the loop of a single interest.

Also, although not proven by results, the proposed approach is designed theoretically to perform on concise semantic ontologies. Consequently, scaling the KB to the size of a complete LoD KB like DBPedia or Freebase would not be feasible and the runtime would not be acceptable at least not on an ordinary processor.

In addition, the implementation presented in this thesis suffers some technical limitations which could be solved in future releases. First of those limitations is the complete dependence on the SMILE library to perform the BN related tasks which leaves the system with very little control over the BN and the inference algorithms. The second limitation is that the context selection in this implementation is performed manually by the user. More importantly, the evaluations presented in this thesis failed to test a lot of important aspects due to lack of data needed for such tests. For example, the tests evaluated whether the system performs what it theoretically should do, i.e. recommend videos that are most related to the ones that were observed to have positive feedback from the user. However, the tests failed to evaluate whether the recommendations really were interesting to the user.

6.2 Future Work

Judging by the promising results obtained from the proposed approach, further development of this idea may indeed lead to a very powerful RS. As a result, driven by its limitations, the work presented in this thesis opens several rooms for improvement.

From a conceptual point of view, future research in this topic may address the different methods to increase the expressiveness of the proposed model so that OWL ontologies including relation constrained classes can be completely transformed into BNs without losing any information. In that manner, the relations between the nodes of the BN would be established in a better way, and consequently, the quality of the recommendations would improve. In addition, Machine Learning techniques can also be applied to develop methods, which automatically assign the weight of the dependencies between the nodes, instead of the static approach that was used in this thesis.

Furthermore, the temporal aspect could be added to the model by introducing some temporal decay of interests which weighs the older evidences with less weights than the more recent ones. Or a possible alternative is to convert the simple BN model to a DBN in order to fuse in the temporal information in the model without applying major conceptual changes in the methodology. In the same sense, investigations about how to alter the model to give bias to the more recent videos could be performed. Equally important, a mechanism which automatically detects the context in which the user currently is either by supervised training or by pattern disruption detection could be developed.

Additionally, the implementation could be improved by using a more flexible Bayesian inference library or even developing a new one from scratch which will provide more control

over the network and the inference algorithms and consequently performance trade-offs can be performed. Also the implementation of the model can be further extended to allow more values/states per node instead of just two. In that manner, the system can function with ratings instead of just binary values.

An essential point in every system is testing and evaluation. Therefore, the proposed approach needs to undergo a set of well designed user tests to evaluate how accurate the interests of the user are captured and consequently enhance the approach and increase its capabilities in generating more relevant recommendations. In order to be able carry out this step fairly, techniques for accurately annotating the videos need to be further developed to ensure that the input to the system is semantically correct with respect to the quality of the annotations and the confidence in the weights.

Last but not least, more work can be directed towards adding extra features to the model in order to offer more sophisticated results. Ideas for such features include, but are not limited to, further development of the group recommendations model and adding the support for recommendation diversification.

Bibliography

[1] Sofiane Abbar, Mokrane Bouzeghoub, and Stéphane Lopez. Context-Aware Recommender Systems: A Service-Oriented Approach. In *VLDB PersDB Workshop*, pages 1–6, 2009.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of Recommender Systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 217–253. Springer US, 2011.

[4] Akram Al-Kouz. *Interests Discovery in Social Networks Based on a Semantically Enriched Bayesian Network Model*. PhD thesis, Fakultät IV – Elektrotechnik und Informatik – der Technischen Universität Berlin, 2013.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[6] Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook : Theory, implementation, and applications*. Cambridge University Press, 2003.

[7] Marko Balabanović and Yoav Shoham. Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[8] Jie Bao, Elisa F. Kendall, Deborah L. McGuinness, and Peter F. Patel-Schneider. OWL 2 Web Ontology Language Quick Reference Guide (Second Edition). Technical Report, OWL Working Group, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/.

[9] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. RDF 1.1 Turtle. Technical Report, RDF Working Group, W3C, February 2014. http://www.w3.org/TR/turtle/.

[10] Tim Berners-Lee. *Linked Data*, 2006 (retrieved September 3, 2014). http://www.w3.org/DesignIssues/LinkedData.html.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[12] Dan Brickley and R.V. Guha. RDF Schema 1.1. Technical Report, RDF Working Group, W3C, February 2014. http://www.w3.org/TR/rdf-schema/.

[13] Huajun Chen, Zhaohui Wu, and Philippe Cudré-Mauroux. Semantic Web Meets Computational. *IEEE Computational Intelligence Magazine*, 7(2):67–74, 2012.

[14] Ruey-Shun Chen, Yung-Shun Tsai, K. C. Yeh, D. H. Yu, and Yip Bak-Sau. Using Data Mining to Provide Recommendation Service. *WSEAS Transactions on Information Science and Applications*, 5(4):459–474, 2008.

[15] LinkedTV Consortium. *LinkedTV platform*, 2012 (retrieved September 6, 2014). http://www.linkedtv.eu/development/linkedtv-platform/.

[16] World Wide Web Consortium. *W3C Semantic Web Activity*, 2013 (retrieved September 3, 2014). http://www.w3.org/2001/sw/.

[17] Gregory F. Cooper. The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[18] Richard Cyganiak and Anja Jentzsch. *Linking Open Data Cloud Diagram*, 2011 (retrieved September 3, 2014). http://lod-cloud.net/.

[19] Paul Dagum and Michael Luby. Approximating Probabilistic Inference in Bayesian Belief Networks. *Artificial Intelligence*, 60(1):141–153, 1993.

[20] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The YouTube Video Recommendation System. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM, ACM.

[21] L.M. de Campos, J.M. Fernandez-Luna, J.F. Huete, and M.A Rueda-Morales. Group Recommending: A methodological approach based on Bayesian Networks. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 835–844, 2007.

[22] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Combining Content-based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. *International Journal of Approximate Reasoning*, 51(7):785–799, 2010.

[23] Thomas L Dean and Keiji Kanazawa. Probabilistic temporal reasoning. In *Proceedings of AAAI-88*, pages 524–529, 1988.

[24] Tommaso Di Noia, Iván Cantador, and Vito Claudio Ostuni. Linked Open Data-enabled Recommender Systems: ESWC 2014 Challenge on Book Recommendation. pages 129–143, 2014.

[25] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8, New York, NY, USA, 2012. ACM.

[26] Sebastian Faubel. *W3C Semantic Web Layers*, 2007 (retrieved October 10, 2014). http://en.wikipedia.org/wiki/File:W3c-semantic-web-layers.svg.

[27] Robert M. Fung and Kuo-Chu Chang. Weighing and Integrating Evidence for Stochastic Simulation in Bayesian Networks. In *Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '89, pages 209–220. North-Holland Publishing Co., 1990.

[28] Fabien Gandon and Guus Schreiber. RDF 1.1 XML Syntax. Technical Report, RDF Working Group, W3C, February 2014. http://www.w3.org/TR/rdf-syntax-grammar/.

[29] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.

[30] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an Ontology? In *Handbook on Ontologies*, pages 1–17. Springer Berlin Heidelberg, 2009.

[31] Haipeng Guo and William Hsu. A Survey of Algorithms for Real-Time Bayesian Network Inference. In *The joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.

[32] M. Henrion. Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. *Uncertainty in Artificial Intelligence*, 2:317–324, 1988.

[33] Félix Hernández del Olmo and Elena Gaudioso. Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35(3):790–804, 2008.

[34] J.B. Hey. System and method for recommending items, February 26 1991. US Patent 4,996,642.

[35] Rüdiger Klein, Manuel Kober, Dorothea Tsatsou, Tomáš Kliegr, Jaroslav Kuchař, Maria Loli, Vasileios Mezaris, Matei Mancas, Julien Leroi, and Lyndon Nixon. D4.1 Specification of User Profiling and Contextualization. Technical Report, Fraunhofer IAIS, CERTH, UMONS, UEP and STI, 2012.

[36] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[37] Alexander V. Kozlof and Jaswinder P. Singh. A parallel Lauritzen-Spiegelhalter algorithm for probabilistic inference. In *Proceedings of the 1994 ACM/IEEE Conference on Supercomputing*, Supercomputing '94, pages 320–329. IEEE Computer Society Press, 1994.

[38] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.

[39] S. L. Lauritzen and D. J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[40] Adam Lella. January 2014 US Online Video Rankings, February 2014. https://www.comscore.com/Insights/Press-Releases/2014/2/comScore-Releases-January-2014-US-Online-Video-Rankings.

[41] Greg Linden, Brent Smith, and Jeremy York. Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[42] D. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2004.

[43] Brian McBride. RDF Primer. Technical Report, RDF Core Working Group, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[44] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical Report, Web Ontology Working Group, W3C, February 2004. http://www.w3.org/TR/owl-features/.

[45] David Nadeau and Satoshi Sekine. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[46] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. Polylens: A recommender system for groups of users. In *ECSCW 2001*, pages 199–218. Springer Netherlands, 2001.

[47] Nikolaos Pappas and Andrei Popescu-Belis. Combining Content with User Preferences for TED Lecture Recommendation. In *11th International Workshop on Content Based Multimedia Indexing*. IEEE, 2013.

[48] Nikolaos Pappas and Andrei Popescu-Belis. Sentiment Analysis of User Comments for One-Class Collaborative Filtering over TED Talks. In *36th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2013.

[49] Judea Pearl. Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence*, 29(3):241 – 288, 1986.

[50] Judea Pearl and Stuart Russell. Bayesian networks. Technical report, Handbook of Brain Theory and Neural Networks, 1995.

[51] Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, and Ognyanoff Miroslav Goranov. Towards Semantic Web Information Extraction. In *Proceedings of ISWC (Sundial Resort)*, 2003.

[52] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. Technical Report, RDF Data Access Working Group, W3C, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[53] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[54] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to Recommender Systems Handbook*. Springer, 2011.

[55] Giuseppe Rizzo and Raphaël Troncy. Nerd: A framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 73–76, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[56] Balthasar Schopman, Dan Brickly, Lora Aroyo, Chris van Aart, Vicky Buser, Ronald Siebes, Lyndon Nixon, Libby Miller, Veronique Malaise, Michele Minno, Michele Mostarda, Davide Palmisano, and Yves Raimond. NoTube: Making the Web part of personalised TV. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, 2010.

[57] Guy Shani and Asela Gunawardana. Evaluating recommender systems. Technical Report MSR-TR-2009-159, 2009.

[58] John F. Sowa. Semantic Networks. *Encyclopedia of Artificial Intelligence*, 1987.

[59] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

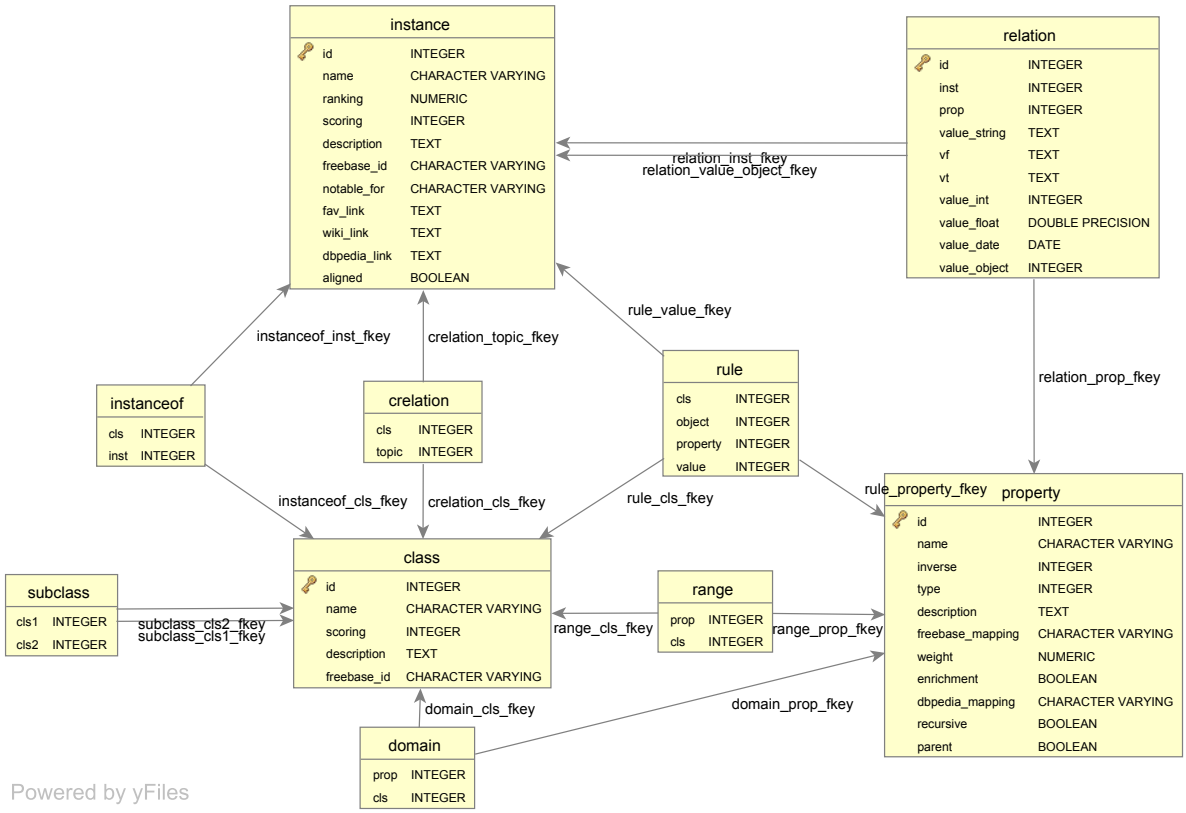
[60] Amrapali Zaveri, Dimitris Kontokostas, Mohamed A. Sherif, Lorenz Bühmann, Mohamed Morsey, Sören Auer, and Jens Lehmann. User-driven quality evaluation of dbpedia. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 97–104, New York, NY, USA, 2013. ACM.

THIS PAGE IS INTENTIONALLY LEFT BLANK

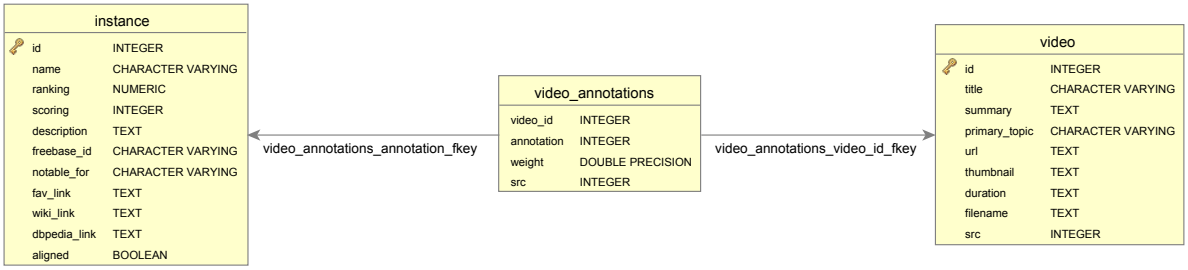
Appendix A

Data Layer

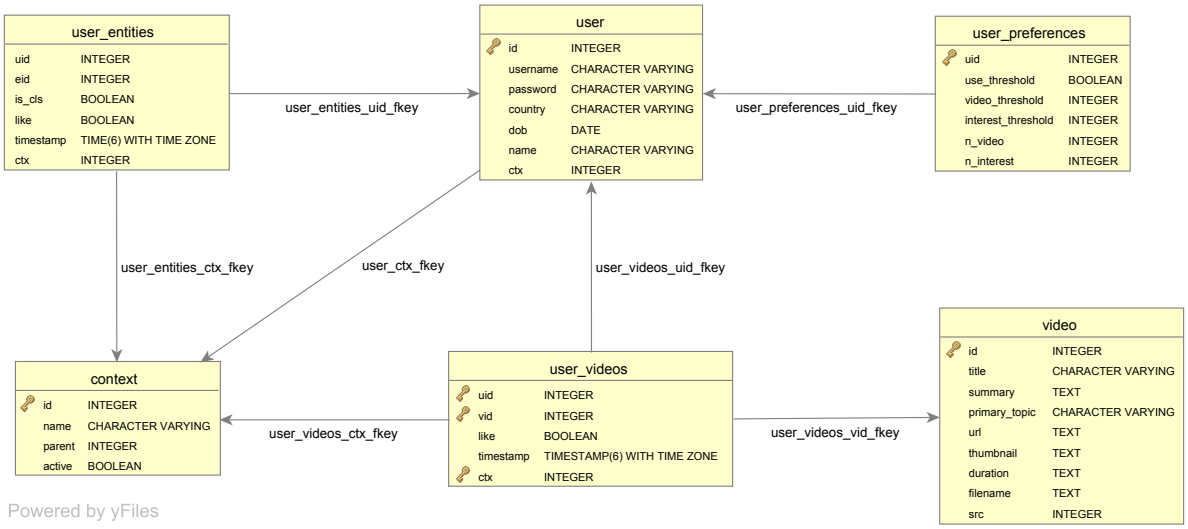
A.1 Knowledge Database Schema



A.2 Video Database Schema



A.3 User Database Schema



Appendix B

Application In Action

B.1 Web Application Features

The BayRec system interface starts as usual with a login page, as shown in Figure B.1, in which the users enter their credentials in order to load their data from the server. After logging in, the user is directed to the home page shown in Figure B.2. In this home page, highlights of the recommended videos and interests are shown.

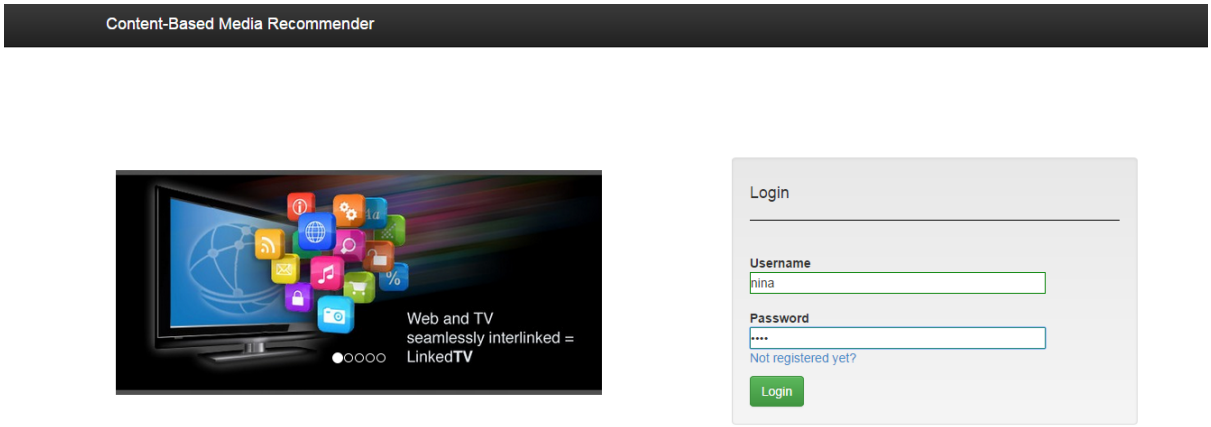


Figure B.1: Login page

Using the interface shown in Figure B.3, the user can utilize the browsing feature to search for videos. The available search options are the substring search in the title of the video or the summary text, search by annotation or search by video topic.

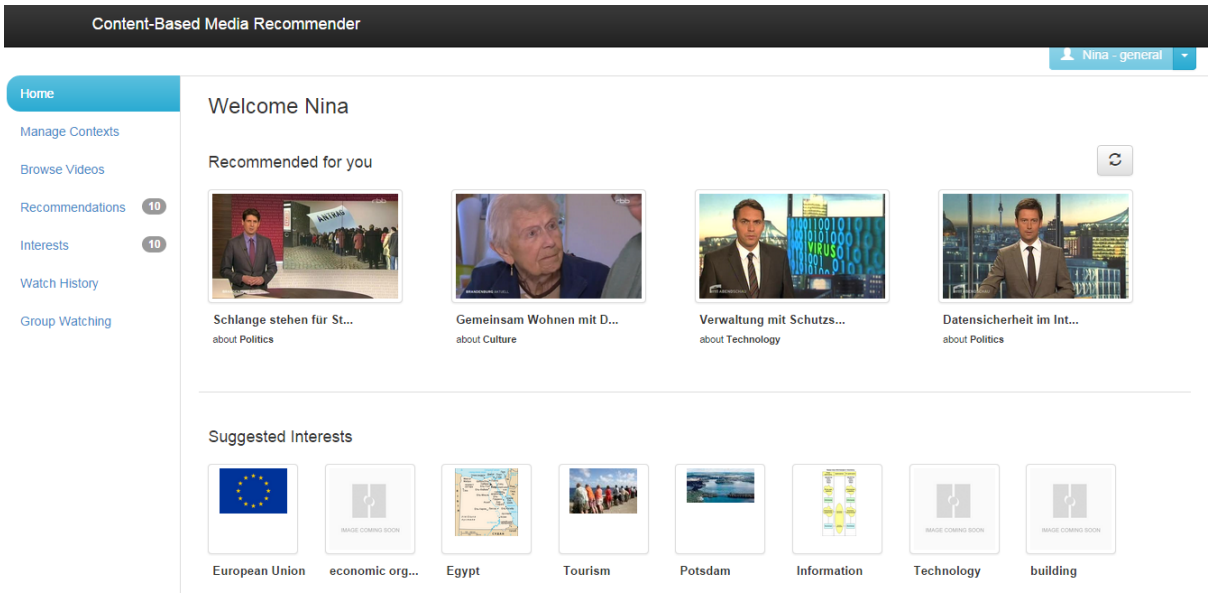


Figure B.2: User homepage showing recommendations and suggestions overview

From the preferences editor feature shown in Figure B.4, the user can control the mode in which the recommendations are generated. The options available in the current version are threshold recommendations and top-N recommendations where the user can control the threshold and the N values.

In addition, users are able to generate recommendations for a group by entering the group members credentials as shown in Figure B.5. The system would then load the user models of each member and generate a group model merging them together and generate the recommendations accordingly.

A key feature in the BayRec system is that users can create and switch between contexts. This feature, shown in Figure B.6a enables the users to control the currently active interests and accordingly improves the quality of the recommendations generated w.r.t. the current time and place.

An additional feature available only in the admin account is the network visualization feature. In this view shown in Figure B.6b, the BN graph is visualized using a force directed graph creation library[1]. The video nodes are represented with orange nodes, instances are represented by light blue nodes and classes by dark blue nodes. The main reason why this feature was implemented is network creation and connection debugging.

[1] http://bl.ocks.org/mbostock/4062045

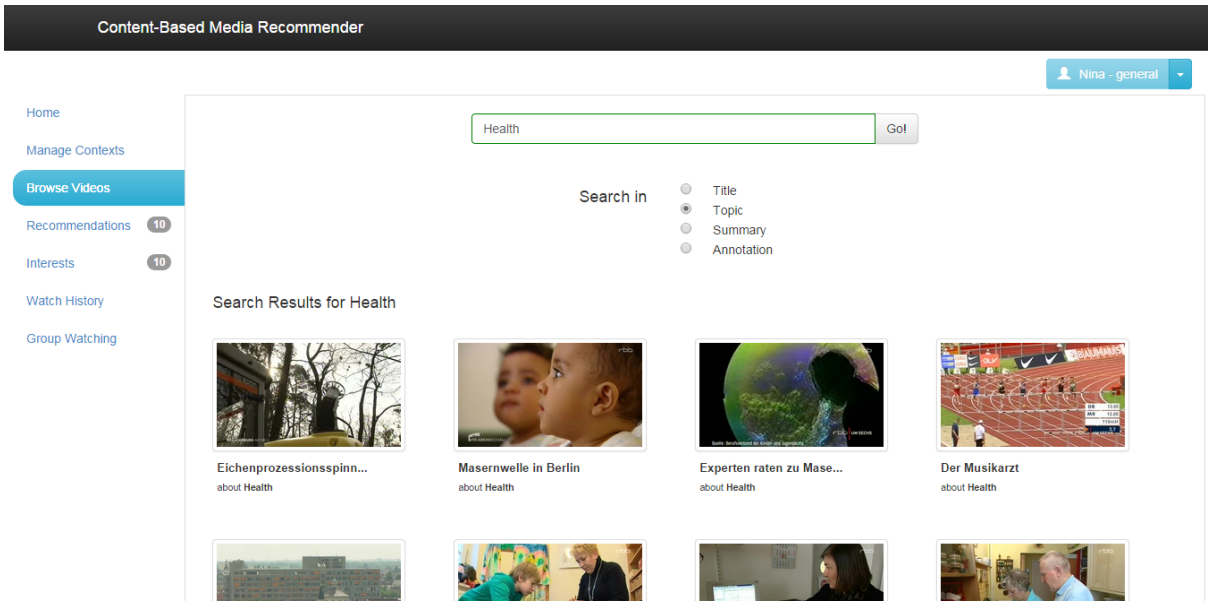


Figure B.3: Video Search by title, topic, summary or annotation

B.2 Recommendation Examples

In this section, an example for the change in recommendations based on the watch history is shown. The user Nina enters her username and password and logs in. Initially, the video recommendations page looks as shown in Figure B.7. The videos have around 50% interest range since no information about the user interests are available initially. The same holds for the interests page shown in Figure B.8.

Using the browsing feature, the Nina starts searching for videos of interest to her. She watches the list of videos whose annotations are shown in Table B.1 and gives positive feedback about them. Studying the list of watched videos, it can be observed that Nina seems to be interested in Measles and Cancer related events taking place in Germany.

Nina now decides to see what the system recommends for her to watch next. The results of the updated recommendations and interests are shown in Figures B.9 and B.10 respectively. Notice in Figure B.10 how the interests list is updated to all Medicine and Disease related topics. Table B.2 shows the detailed annotations of the top-5 recommended videos in the recommendations list. It can be observed that the videos annotated with topics most similar to the ones already viewed are recommended first to Nina like the first three videos. Also notice how the system smartly recommends other videos about Diseases and Medicine although the annotations Disease and Medicine never explicitly appeared in Nina's watch list. However, using semantics, the system was able to learn that Nina is for example interested in Disease topics since both Measles and Cancer are diseases.

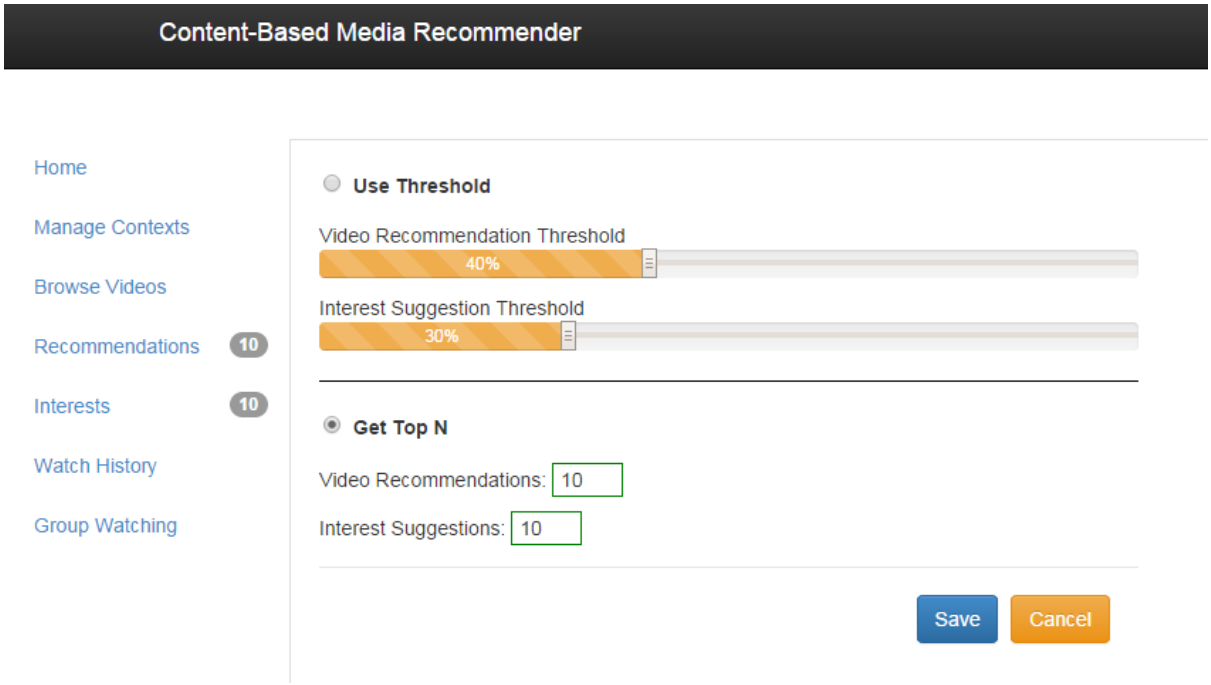


Figure B.4: Controlling recommendation preferences

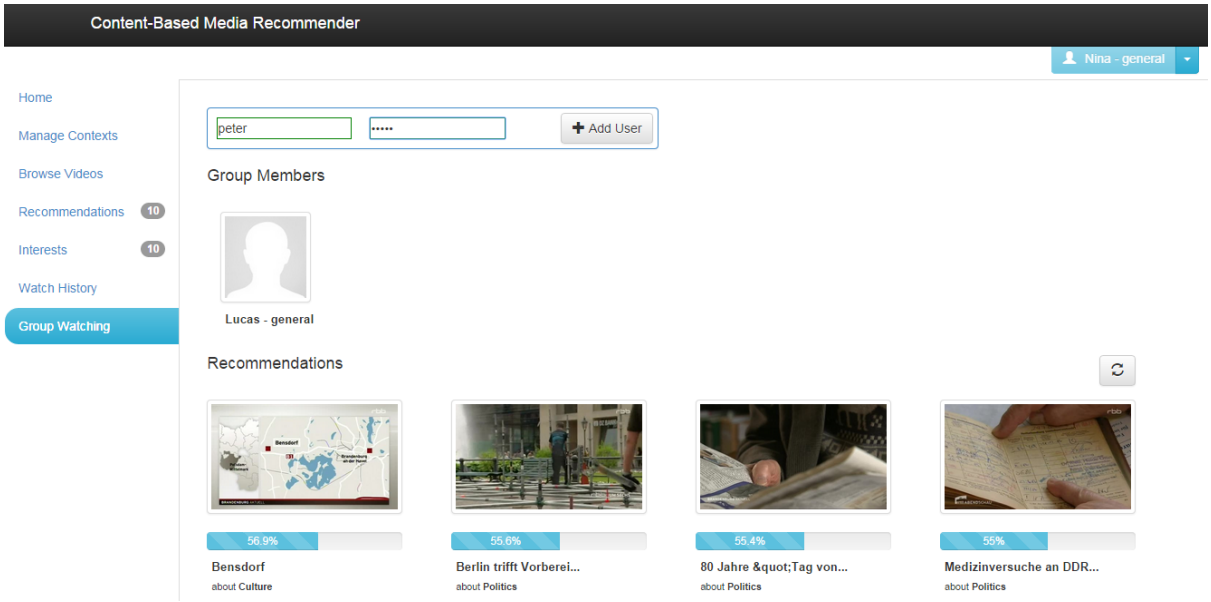
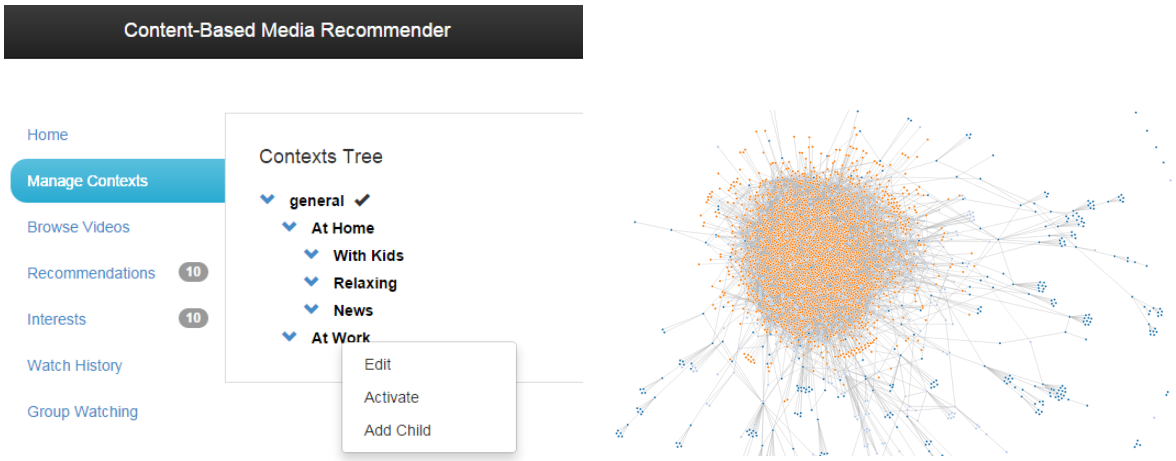


Figure B.5: Recommendations for a group



(a) Context Management Tree (b) Bayesian Network visualized as a graph

Figure B.6: Some of the BayRec system features

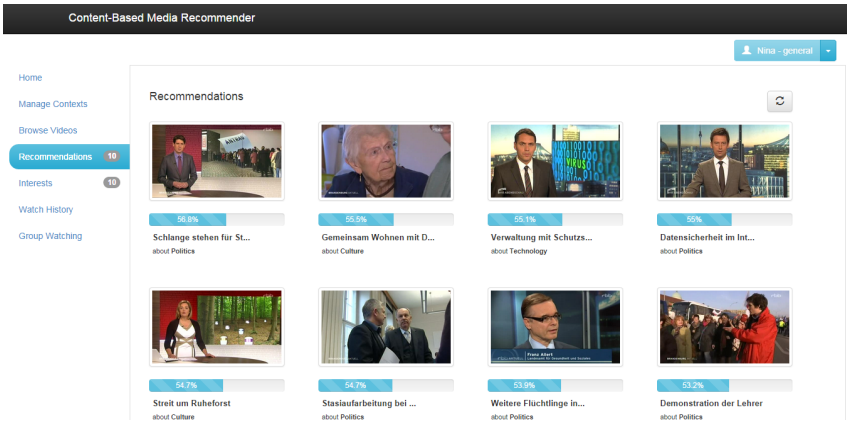


Figure B.7: The initial recommendations before showing interest in any topics or videos

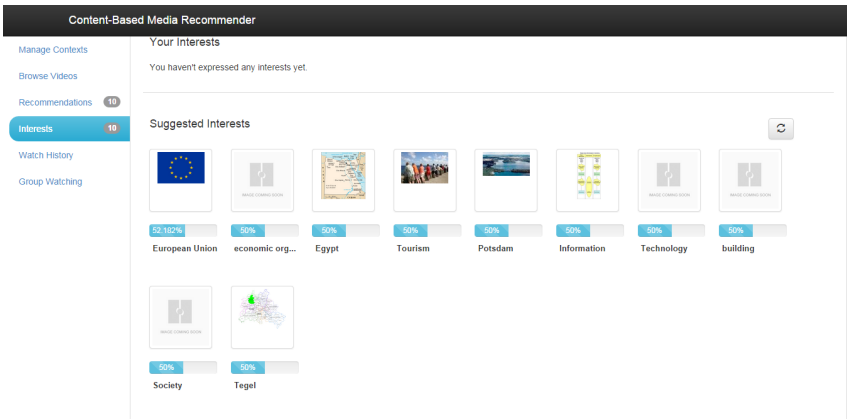


Figure B.8: The initial interests before showing interest in any topics or videos

Video	Content	Weight
Experten raten zu Masernimpfung	Measles	0.95
	Berlin	0.77
	Vaccination	0.76
	Conjunctivitis	0.72
	Pediatric	0.61
Immer mehr Masern-Erkrankungen	Measles	1.0
	Conjunctivitis	0.87
	Vaccination	0.84
	Berlin	0.76
	Infection	0.69
Bundesweites Krebsregister wird eingerichtet	Chemotherapy	1.0
	Breast Cancer	0.94
	German Democratic Republic	0.83
	Cancer	0.82
	Brandenburg	0.76
Krebskongress in Potsdam	Chemotherapy	1.0
	Breast Cancer	0.99
	Oncology	0.90
	Brannenburg	0.88
	Cancer	0.81

Table B.1: List of annotated videos watched with positive feedback

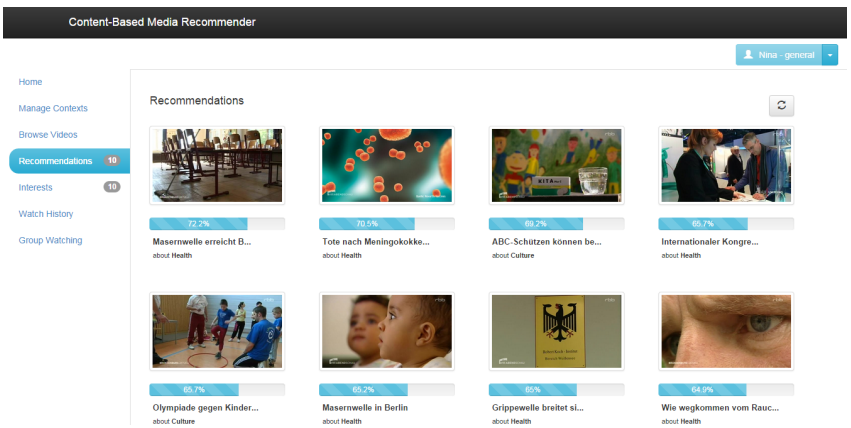


Figure B.9: The recommendations after showing interest in some videos

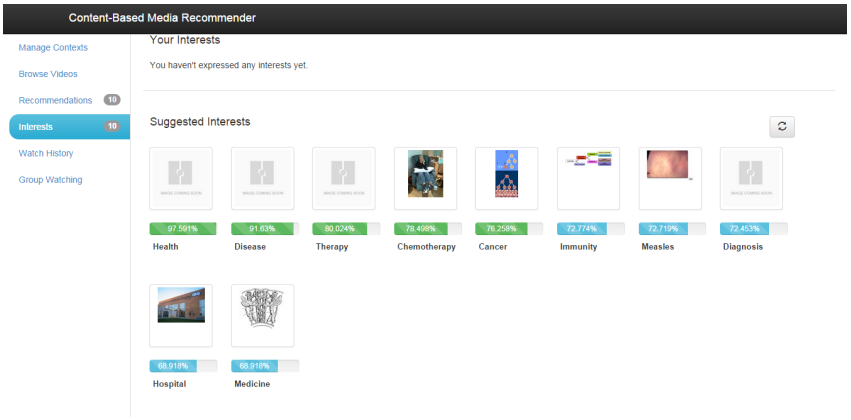


Figure B.10: The interests after showing interest in a some videos

Video	Content	Weight
Masernwelle erreicht Brandenburg	Measles	1.0
	Pneumonia	0.80
	Vaccination	0.78
Tote nach Meningokokken-Infektion	Berlin	0.78
	Infection	0.66
	Vaccination	0.47
	Immunity	0.45
	Disease	0.43
ABC-Schützen können besser Deutsch	Measles	0.88
	Berlin	0.77
	Vaccination	0.61
	Turkey	0.465
Internationaler Kongress für Gefäßmediziner	Prague	0.83
	Leipzig	0.82
	Medicine	0.64
	Disease	0.32
Olympiade gegen Kinder-Speck	Berlin	0.73
	Potsdam	0.72
	Disease	0.51

Table B.2: The top-5 recommended videos (annotated)