

TKK Dissertations 227
Espoo 2010

**CRISP, FUZZY, AND PROBABILISTIC FACETED
SEMANTIC SEARCH**

Doctoral Dissertation

Markus Holi



**Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology**

TKK Dissertations 227
Espoo 2010

CRISP, FUZZY, AND PROBABILISTIC FACETED SEMANTIC SEARCH

Doctoral Dissertation

Markus Holi

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium AS1 at the Aalto University School of Science and Technology (Espoo, Finland) on the 9th of June 2010 at 12 noon.

**Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology**

**Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Mediatekniikan laitos**

Distribution:

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology
P.O. Box 15500 (Otaniementie 17)
FI - 00076 Aalto
FINLAND
URL: <http://media.tkk.fi/>
Tel. +358-9-47001
E-mail: markus.holi@metropolia.fi

© 2010 Markus Holi

ISBN 978-952-60-3183-5
ISBN 978-952-60-3184-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: <http://lib.tkk.fi/Diss/2010/isbn9789526031842/>

TKK-DISS-2765

Multiprint Oy
Espoo 2010

ABSTRACT OF DOCTORAL DISSERTATION		AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY P.O. BOX 11000, FI-00076 AALTO http://www.aalto.fi	
Author Markus Holi			
Name of the dissertation Crisp, Fuzzy, and Probabilistic Faceted Semantic Search			
Manuscript submitted 21.09.2009		Manuscript revised 09.05.2010	
Date of the defence 09.06.2010			
<input checked="" type="checkbox"/> Monograph		<input type="checkbox"/> Article dissertation (summary + original articles)	
Faculty	Faculty of Information and Natural Sciences		
Department	Department of Media Technology		
Field of research	Media Technology		
Opponent(s)	Dr. Jacco van Ossenbruggen and Dr. Ossi Nykänen		
Supervisor	Prof. Eero Hyvönen		
Instructor	Prof. Eero Hyvönen		
<p>Abstract</p> <p>This dissertation presents contributions to the development of the <i>faceted semantic search (FSS)</i> paradigm. First, two fundamental solutions to <i>FSS</i>, which have been widely used since their development are presented. The first is the projection of search facets from annotation ontologies using logical rules. The second is the logic rule-based generation of recommendation links for search items based on the semantic relations of these items.</p> <p>After presenting these solutions, the rest of the dissertation focuses on solving the following deficiencies of <i>FSS</i>: the lack of capabilities to model uncertainty, the inability to rank search results according to relevance, and the usability problems resulting from naively using annotation ontology concepts as search categories. Two sets of solutions to these problems are presented.</p> <p>First, a <i>fuzzy faceted semantic search (FFSS)</i> framework is developed, which extends the crisp set basis of <i>FSS</i> to fuzzy sets. This framework is based on two main ingredients: First, weighted annotations, which are used to determine the membership degrees of search items in annotation concepts. Second, fuzzy mappings of separate end-user categories onto the annotation concepts.</p> <p>In addition, also a <i>probabilistic faceted semantic search (PFSS)</i> framework was developed, which incorporates weighted annotations, modeling of uncertainty in Semantic Web taxonomies, sophisticated mappings of end-user facets onto annotation ontologies, and the combination of evidence from multiple ranking schemes.</p> <p>These ranking methods were empirically analyzed. According to the preliminary evaluation both ranking methods significantly improve quality of search results compared to crisp <i>FSS</i>. Both also outperformed a currently used heuristical ranking method. However, in the case of <i>FFSS</i> this difference did not reach the level of statistical significance.</p>			
Keywords Semantic Web, Ontology, Fuzzy sets, Probability theory, Faceted Search			
ISBN (printed) 978-952-60-3183-5		ISSN (printed) 1795-2239	
ISBN (pdf) 978-952-60-3184-2		ISSN (pdf) 1795-4584	
Language English		Number of pages 216+9	
Publisher Aalto University School of Science and Technology			
Print distribution Aalto University School of Science and Technology			
<input checked="" type="checkbox"/> The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789526031842/			

VÄITÖSKIRJAN TIIVISTELMÄ		AALTO-YLIOPISTO TEKNILLINEN KORKEAKOULU PL 11000, 00076 AALTO http://www.aalto.fi	
Tekijä Markus Holi			
Väitöskirjan nimi Täsmällinen, sumea ja probabilistinen semanttinen näkymähaku			
Käsikirjoituksen päivämäärä	21.09.2009	Korjatun käsikirjoituksen päivämäärä	09.05.2010
Väitöstilaisuuden ajankohta 09.06.2010			
<input checked="" type="checkbox"/> Monografia		<input type="checkbox"/> Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)	
Tiedekunta	Informaatio- ja luonnontieteellinen tiedekunta		
Laitos	Mediatekniikan laitos		
Tutkimusala	Viestintäteknikka		
Vastaväittäjä(t)	Dr. Jacco van Ossenbruggen ja TKT Ossi Nykänen		
Työn valvoja	Prof. Eero Hyvönen		
Työn ohjaaja	Prof. Eero Hyvönen		
<p>Tiivistelmä</p> <p>Tämä väitöskirja esittelee kontribuutioita semanttisen moninäkömahaun kehitykselle. Aluksi esitellään kaksi perustavanlaatuaista ja semanttisen moninäkömahaun piirissä laajalti käytettyä menetelmää. Ensimmäinen näistä on näkymien projisointi annotointitontologioista loogisten sääntöjen avulla. Toinen on semanttisiin suhteisiin ja loogisiin sääntöihin perustuva suosittelulinkkien generointi hakutuloksille.</p> <p>Näiden menetelmien esittelyn jälkeen keskitytään kolmeen semanttisen moninäkömahaun ongelmaan: 1) menetelmän puute epätäsmällisen tiedon mallintamiseen, 2) menetelmän puute tietojen järjestämiseen relevanssin perusteella, 3) käytettävyysongelmat, jotka ovat seurausta annotointitontologioiden käsitteiden suoraviivaisesta käyttämisestä hakukategorioina. Näihin ongelmiin esitetään kaksi eri ratkaisutapaa.</p> <p>Ensin esitellään sumea semanttinen moninäkömähaku (SSM), joka laajentaa semanttisen moninäkömahaun joukko-opillista perustaa sumeisiin joukkoihin. SSM hyödyntää painotettuja annotointeja, joiden avulla määritellään haun kohteiden kuuluminen annotointikäsitteisiin. Lisäksi SSM:ssä näkymät määritellään erillään annotointitontologioista ja hakukategoriat peilataan sumeasti annotointitontologioiden käsitteille.</p> <p>Sumeisiin joukkoihin perustuvan ratkaisutavan jälkeen esitellään probabilistinen semanttinen moninäkömähaku (PSM), joka sisältää painotetut annotoinnit, epätäsmälliset käsittehierarkiat, hakukategorioiden hienostuneet peilaukset annotointikäsitteille, sekä relevanssia koskevan evidenssin yhdistämisen useasta eri algoritmista.</p> <p>Lopuksi sumeaa ja probabilistista semanttista moninäkömähakua arvoitetaan niiden tulostenjärjestämiskyvyn perusteella. Alustavan arvioinnin perusteella voidaan todeta, että kumpikin menetelmä järjestää hakutuloksia relevanssin perusteella paremmin kuin perinteinen—täsmällinen—semanttinen moninäkömähaku. Kumpikin myös järjestää tuloksia paremmin kuin nykyisin joissakin semanttisissa moninäkömähakusovelluksissa käytössä oleva heuristinen järjestysalgoritmi, tosin SSM:n tapauksessa ero ei ollut tilastollisesti merkitsevä.</p>			
Asiasanat Semanttinen verkko, ontologiat, sumea logiikka, todennäköisyyslaskenta, näkömähaku			
ISBN (painettu)	978-952-60-3183-5	ISSN (painettu)	1795-2239
ISBN (pdf)	978-952-60-3184-2	ISSN (pdf)	1795-4584
Kieli	englanti	Sivumäärä	216+9
Julkaisija Aalto-yliopiston teknillinen korkeakoulu			
Painetun väitöskirjan jakelu Aalto-yliopiston teknillinen korkeakoulu			
<input checked="" type="checkbox"/> Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789526031842/			

Preface

Work related to this dissertation began in 2003 when I started to work at the Semantic Computing (SeCo) research group, and was introduced to the fascinating world of the Semantic Web.

I would like to express my warmest thanks to Professor Eero Hyvönen, the leader of SeCo and the supervisor of this dissertation for his valuable guidance and encouragement I have received during the years. Big thanks also to Kim Viljanen and Petri Lindgren, the other co-authors of the articles that form the basis of large parts of this dissertation. I am also indebted to many other researchers at SeCo for the inspiring and rewarding collaboration.

I am deeply thankful to my current employer Metropolia University of Applied Science for being supportive and understanding toward this work. I would also like to thank my collaborators in the projects I have participated at Metropolia. This inspiring collaboration has contributed to this dissertation in many ways.

This work has been funded partly by TEKES - the Finnish Funding Agency for Technology and Innovations. The work has been conducted at and using the resources of the University of Helsinki, TKK, and Metropolia.

I feel indebted to those who give me the broader context in life. Thanks to my parents, brothers, and sisters for everything. And finally, I want to thank my wife Hanna and my children Alisa, Elias, Sofia, and Daniel for their continuous support. You are truly a source of energy and a motivating factor in my life.

Mäntsälä, 09.05.2010 Markus Holi

Contents

Preface	7
Contents	9
List of Figures	15
List of Tables	19
1 Introduction	21
1.1 Faceted Search	21
1.2 Faceted Semantic Search	23
1.3 Faceted Semantic Search and Uncertainty	24
1.4 Problems and Solutions	25
1.5 Organization of the Dissertation	29
I CRISP FACETED SEMANTIC SEARCH	31
2 Introduction to Faceted Semantic Search	32
2.1 How Faceted Search Works?	32
2.2 Set-theoretic Interpretation of Faceted Search	36
2.3 Facets and Metadata	37
2.4 The Promise of Faceted Semantic Search	39
2.5 Content Creation for Faceted Semantic Search	40
2.5.1 Ontology Creation and Reuse	40
2.5.2 Annotation of Content	43
2.5.3 Attachment of Uncertain Information to Ontologies and An- notations	45
2.5.4 Evolution of Content	46

3	Designing and Creating a Website Based on RDF Content	49
3.1	Two Views of the Semantic Web	49
3.2	Example Applications	51
3.3	Specifying the Transformation	54
3.4	Website Generation	60
3.5	Related Work	63
3.6	Conclusions	64
3.7	Contributions and Significance for Later Research	66
4	Problems of Crisp Faceted Semantic Search	67
4.1	Lack of Capability to Model Uncertainty	67
4.2	Information Overflow: Lack of Ranking	68
4.3	Lack of User-Centric Facets	69
II	FUZZY APPROACH	71
5	Introduction to the Fuzzy Approach	72
5.1	Crisp Sets	73
5.2	Fuzzy Sets	74
6	Fuzzy Faceted Semantic Search	77
6.1	Architecture of the Framework	78
6.2	Fuzzy Annotations	80
6.3	Fuzzy Mappings	82
6.4	Mappings to Boolean Concepts	86
6.5	Performing the Search	89
6.6	Implementation	90
6.6.1	Representing Fuzzy Annotations	90
6.6.2	Representing Search Facets	91
6.6.3	Projection of Annotations	92
6.7	Related Work	93

6.8	Summary	94
7	Integrating Term Frequency - Inverse Document Frequency (TF-IDF) Weighting with Fuzzy Faceted Semantic Search	96
7.1	The TF-IDF Method	97
7.2	Ontological Extension of TF-IDF	98
7.3	Evaluation	100
8	Contributions and Lessons Learned	102
8.1	Contributions	102
8.2	Lessons Learned	104
III	PROBABILISTIC APPROACH	107
9	Introduction to the Probabilistic Approach	108
9.1	Basic Probability Theory	111
9.2	Bayesian Networks	114
10	Modeling Degrees of Overlap Between Concepts in Semantic Web Taxonomies	118
10.1	The Problem and the Solution Approach	118
10.2	Representing Overlap	123
10.3	Solid Path Structure	125
10.4	Computing the Overlaps	128
10.4.1	The Solid Path Structure Approach	128
10.4.2	The Bayesian Network Approach	129
10.5	Implementation	133
10.5.1	Overlap Graph	133
10.5.2	Overlap Computations	133
10.6	Conclusions	135
10.7	Related Work	137

11 Probabilistic Faceted Semantic Search	140
11.1 A Motivating Example	142
11.2 PFSS in a Nutshell	145
11.2.1 The Intuitive Interpretation of <i>PFSS</i>	146
11.2.2 The High-level <i>PFSS</i> Algorithm	148
11.2.3 The <i>PFSS</i> Probability Model	151
11.3 Simple Search Category	154
11.3.1 The Base Case for Documents with a Single Annotation . .	154
11.3.2 The Base Case for Documents with Multiple Annotations . .	158
11.3.3 The Recursion Step	161
11.3.4 The Algorithm	161
11.4 Search Categories Mapped to Boolean Combinations of Annotation Concepts	162
11.4.1 OR	162
11.4.2 AND	165
11.4.3 NOT	167
11.4.4 The Algorithm	169
11.5 Hierarchical Search Categories	170
11.6 Combining Evidence of Multiple Ranking Schemes	172
11.7 Performing the Search	174
11.8 Summary	174
12 Contributions and Lessons Learned	176
12.1 Contributions	176
12.2 Lessons Learned	177
IV EVALUATION AND CONCLUSIONS	181
13 Empirical Comparison of Ranking Methods	182
13.1 HealthFinland — A Semantic Health Portal for the General Public	183
13.2 The Dataset	184

13.3 Interpreting the Data	185
13.3.1 Annotations and Annotation Concepts	185
13.3.2 End-user Facets	186
13.4 Evaluation	188
14 Results and Discussion	194
14.1 Results	194
14.2 Discussion	198
References	202

Appendix A PFSS Implementation

A.1 Semantic Knowledge Base
A.2 PFSS Knowledge Base
A.3 PFSS Creator
A.4 Facet Producer
A.5 Matrix Conceptor
A.6 Mass Calculator
A.7 Annotation Persister
A.8 Document Relevance Computer
A.9 Ranking Scheme
A.10 End-user Facet Creator
A.11 Facet Mapper
A.12 Query Interface
A.13 The <i>PFSS</i> Container

List of Figures

1.1	The timeline of this dissertation work. The work included in this dissertation is in bold, and work influencing or influenced by the dissertation work in normal font.	25
2.1	The start page of the <i>HealthFinland</i> faceted health portal.	32
2.2	The result list page of the <i>HealthFinland</i> faceted health portal for the selection of the search category <i>Smoking</i>	33
2.3	The result list page of the <i>HealthFinland</i> faceted health portal for the search category selection of <i>Smoking</i> and <i>Pregnancy</i>	34
2.4	The document page of the <i>HealthFinland</i> faceted health portal.	35
2.5	Depiction of faceted classification. Search items are classified according to each of the facets.	39
2.6	Content Creation and Evolution	46
3.1	Rendering RDF(S) content as an HTML website.	50
3.2	A photo exhibition generated with SWeHG.	52
3.3	Transforming an RDF repository into HTML pages.	55
3.4	Internal architecture of SWeHG.	60
3.5	An analysis page created by SWeHG.	61
6.1	Components of the fuzzy faceted semantic search (FFSS) framework.	78
6.2	Real-world examples of annotation projection cases	87
9.1	An example of a DAG.	116
10.1	A Venn diagram illustrating countries, areas, their overlap, and size in the world.	119
10.2	A standard Semantic Web taxonomy based on the Venn diagram of Figure 10.1.	120

10.3	Representing Lapland’s overlaps by partitioning it according to the areas it overlaps. Each part is subsumed by both Lapland and the respective country.	121
10.4	The taxonomy corresponding to the Venn diagram of Figure 10.1.	124
10.5	The taxonomy of Figure 10.4 as a solid path structure.	127
10.6	The architecture of the implementation.	134
10.7	Quantification of concepts. The number of direct instances of each concept is 10. In the case of partial subsumption, only a part of the mass of the subconcept is taken as the mass of the superconcept	135
10.8	Overlap graph where mirror is partially overlapping furniture, car parts, and personal belongings	137
10.9	Overlap graph of Figure 10.8 quantified and transformed to solid path structure.	138
11.1	A mockup of the Virtual Mental Health Center portal.	143
11.2	Conceptual structure behind the Virtual Mental Health Center search functionality.	145
11.3	An example screenshot of a simple single faceted search.	155
11.4	The conceptual model of the simple scenario for probabilistic faceted search.	156
11.5	Multiply annotated document.	158
11.6	The search category <i>Social Problems</i> is defined using the Boolean combination <i>OR</i> of annotation concepts <i>F40.1 Social Phobia</i> and <i>F60-69 Personality Disorders</i>	163
11.7	The search category <i>Finnish Fears</i> is defined using the Boolean combination <i>AND</i> of annotation concepts <i>Finland</i> and <i>F40 Phobias</i>	166
11.8	The search category <i>Slight Problems</i> uses <i>OR</i> , <i>AND</i> , and <i>NOT</i> Boolean combinations of annotation concepts in its definition.	168
11.9	Hierarchical organization of search categories.	171
13.1	The average R-Precision values for the four algorithms compared.	190

A.1	The logical architecture of the <i>PFSS</i> framework
A.2	PFDO with simple instance examples
A.3	The <i>PFSS</i> Knowledge base structure

List of Tables

1.1	The problems and solutions presented in this dissertation work. . .	26
5.1	The problems addressed and the solutions provided by <i>fuzzy FSS</i> . .	73
9.1	The problems and solutions presented in this part of the dissertation.	108
9.2	An example conditional probability table for variable C of DAG of Figure 9.1.	116
10.1	The <i>overlap table</i> of Lapland according to Figure 10.1.	122
10.2	The conditional probability table for the random variable EU' along with the verbal interpretation of each case.	132
11.1	The conditional probability table for the document variable D_5 , ac- cording to the annotation of Figure 11.5.	159
13.1	Results of the Bayesian Statistical Comparison of Ranking Methods.	191

1 Introduction

One of the key promises of the Semantic Web [23] is to facilitate information finding and search on the Web. In the Semantic Web, content—such as a webpage—is described using concepts from machine readable conceptual models—i.e., ontologies—expressed in standard knowledge representation languages such as RDFS [8] and OWL [6]. This enables better re-use and sharing of content between different Web applications, and also the creation of more intelligent services for the end-user.

This dissertation focuses on *faceted semantic search (FSS)* [54], a field of Semantic Web research which has already matured enough to produce commercial and public Web applications, such as the Finnish health portal HealthFinland¹ [60]. The work presented in this thesis includes the development of some of the fundamental solutions used in *FSS* up to this day. The main focus, however, is on the development of methods to model uncertainty and conceptual overlap in Semantic Web taxonomies, and weighting of annotations based on textual content of documents to equip *FSS* with such features as ranking of search results, and mapping of separately created end-user search facets onto annotation ontologies.

1.1 Faceted Search

Traditionally, information retrieval on the Web has manifested mainly as free-term search performed using a search engine such as Google². Free-term search works well when the user has a well-defined search problem in mind, e.g., if the user searches for information about a specific person, organization, or a subject that he/she is able to characterize with a simple word or phrase. However, often the user does not

¹<http://www.terveysuomi.fi/fi/etusivu>

²<http://www.google.com>

exactly know what he/she is looking for, or wants to get a high-level understanding of a subject, its inner relations, and relations to other subjects. Activity of this kind is called exploratory search [74], and it can be greatly facilitated by providing the user with navigation aids, such as indices, terms related to the chosen subject, and recommendations.

The faceted search paradigm [86, 43] supports exploratory search well. In faceted search the search items are indexed along multiple orthogonal—i.e., independent—taxonomies that are called *facets*. An individual concept that belongs to a facet is called a *search category*. The graphical user interface (GUI) reveals these facets to the user and enables the browsing of the indexed content according to any of them. In addition, the organization and structure of the facets reveal key relations among the different concepts related to the subject of interest. The faceted search paradigm can be further divided into single-faceted search and multi-faceted search. In single-faceted search the user can browse using one facet at a time, as opposed to multi-faceted search which supports simultaneous search category selections from different facets [100].

The faceted search paradigm is based on *facet analysis* [73], a classification scheme introduced in information sciences by S. R. Ranganathan already in the 1930's. From the 1970's on, facet analysis has been applied in information retrieval research, as a basis for search. The focus and goal of faceted search applications such as HIBROWSE [103], Relation Browser [75], Flamenco [43], and mSpace [94] has been to create intuitive, and easy-to-use search interfaces. An excellent survey—unfortunately only in Finnish—of these faceted search systems can be found in [100].

1.2 Faceted Semantic Search

Faceted search is integrated with the notion of ontologies and the Semantic Web [54, 80, 52, 72, 44] to form the *faceted semantic search (FSS)* paradigm. In the context of the Semantic Web, ontologies are machine readable conceptual models represented in one of the *RDF* [9] based ontology languages, such as *RDFS* [8] and *OWL* [6] recommended by the *W3C*³. These languages contain constructs that are typically needed in conceptual models. For example, in *OWL* the concepts of an ontology are defined as instances of *owl:Class*, and a conceptual hierarchy is created using the *rdfs:subClassOf* relation between concepts. These language constructs have predefined semantics so general purpose software libraries such as Jena[3] have been developed to take care of ontological data processing. Web content such as webpages or other documents found on the Web are described or *annotated* using concepts from these ontologies. Together the ontologies and the annotations of Web content form a *semantic knowledge base*. Semantic knowledge bases like this are the basis for *FSS* systems.

The motivation for the integration of *faceted search* with Semantic Web ontologies is that the latter offers a natural basis for the creation of the facets used in the former. Faceted semantic search has been used successfully in a number of semantic portals. Examples of content publishing tools which use the faceted semantic search paradigm are SWeHG—which is one of the results of this dissertation—and OntoViews. The original idea—which was developed in SWeHG and OntoViews—was to create facets algorithmically from a set of underlying ontologies that are used as the basis for annotating search items. This offered an easy and fast way to create the search facets. Furthermore, the mapping of search items onto search facets can be defined using logic rules. This facilitates more "intelligent" semantic search of indirectly related items.

³<http://www.w3c.org>

1.3 Faceted Semantic Search and Uncertainty

However, not before long it was discovered that ontologies used in the annotation of search items are often created primarily for domain specialists, and are not as such suitable to be presented as facets in a portal offered to the layman [46]. In fact, the straightforward algorithmic creation of facets from annotation ontologies sometimes resulted in rather difficult systems from usability point-of-view, and seemed to undermine the main goal of traditional faceted search systems, namely, the creation of intuitive user interfaces for exploratory search. Another problem that was soon discovered was that because the ontology languages of the Semantic Web are based on crisp logic, they are not able to model the uncertainty inherent in our world, and in the faceted search system itself. According to *FSS*, search items are either relevant or non-relevant to a given search, but there are no degrees or probabilities of relevance. Thus, *FSS* lacks the ability to rank results according to relevance. Ranking of search results, however, is seen as a core feature of information retrieval systems [20], and the significance of this feature is emphasized in environments such as national library collections, national museum collections, or the Internet where the amount of searchable information is vast [30]. The main focus of this dissertation is in developing methods to overcome these problems.

The dominant approaches to modeling uncertainty in ontologies and concept taxonomies are fuzzy logic [113, 97, 110, 18, 77] and probability theory [84, 85, 32, 39, 41, 79, 63, 105, 69]. For this reason we chose to base our solutions on these approaches. We created both a fuzzy and a probabilistic version of faceted semantic search, each of which provides solutions to the above presented problems. These approaches are empirically evaluated and compared as part of this dissertation work.

1.4 Problems and Solutions

Table 1.1 outlines the problems and the corresponding solutions presented in this dissertation. The work included in the dissertation is spread over the years 2003 - 2009, and many of the chapters presented here are based on published scientific papers. Table 1.1 specifies the chapters of this dissertation and the corresponding publication—if applicable—in which the solution is presented. A more detailed listing of the results of this dissertation work is presented in Chapter 14. Figure 1.1 gives a timeline visualization of the dissertation, and related work done in the *Semantic Computing (SeCo)*⁴ research group. The work included in this dissertation is written in bold font.

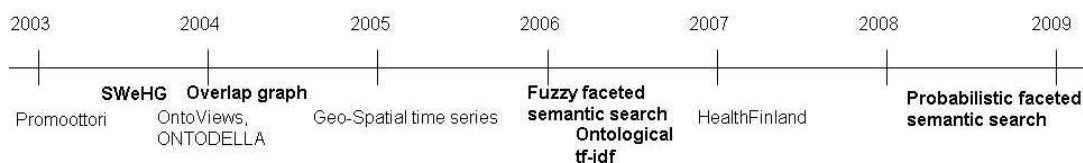


Figure 1.1: The timeline of this dissertation work. The work included in this dissertation is in bold, and work influencing or influenced by the dissertation work in normal font.

The earliest work is presented in Chapter 3, and it dates back to the year 2003, when the *FSS* paradigm itself was still just emerging. Thus, the problems addressed and the solutions developed concentrate on enabling the utilization of Semantic Web ontologies in the creation of an *FSS* system, and in the easy publishing of Semantic Web content to the human user (Problems 1 - 3 in Table 1.1). In short, Chapter 3 presents the projection of facets, and the generation of recommendation links from the ontological knowledge base using logical rules implemented using the the SWI-Prolog programming framework⁵ [13]. These solutions are developed as part

⁴<http://www.seco.tkk.fi>

⁵<http://www.swi-prolog.org>

Table 1.1: The problems and solutions presented in this dissertation work.

	Problem	Solution	Chapter	Publication
1.	How should search facets be created from ontologies to form <i>FSS</i> ?	Logical rule-based projection of facets from ontologies.	3.	Designing and Creating a Web Site Based on RDF Content [49].
2.	How can the ontological knowledge be used to enable semantic browsing of content?	Ontology-based recommendations that are expressed as logical rules.	3.	Designing and Creating a Web Site Based on RDF Content [49].
3.	How can semantic content be published easily on the Web using the ideas of faceted semantic search?	A tool for generating static websites that are organized as simple single-faceted search systems.	3.	Designing and Creating a Web Site Based on RDF Content [49].
4.	Semantic Web ontologies lack the support for modeling uncertainty inherent in the world, including the <i>FSS</i> system itself. This shortcoming hinders the system's ability to provide high quality search results for the user.	A graph notation for representing uncertainty and conceptual overlap in Semantic Web taxonomies, and a Bayesian method for computing degrees of overlap between the concepts of such a taxonomy.	10.	Modeling Uncertainty in Semantic Web Taxonomies [47].
		An ontological extension to the <i>term frequency - inverse document frequency (TF-IDF)</i> method to enable weighting of document annotations based on their textual content.	7.	Integrating TF-IDF Weighting with Fuzzy View-Based Search [48].
5.	Crisp faceted semantic search lacks the capability to rank search results based on relevance.	Fuzzy faceted semantic search.	6.	Fuzzy View-Based Semantic Search [46].
		Probabilistic faceted semantic search.	11.	New work, published in this thesis.
6.	Concepts of annotation ontologies are not always suitable to be presented as search categories on the search GUI.	Fuzzy faceted semantic search.	6.	Fuzzy View-Based Semantic Search [46].
		Probabilistic faceted semantic search.	11.	New work, published in this thesis.
7.	How could rankings of search result provided by different schemes combined to provide better ranking of search results?	Probabilistic faceted semantic search.	11.	New work, published in this thesis.

of a tool—Semantic Web HTML Generator (SWeHG)—which generates a site of static webpages that uses the facet projection solution above to create navigation menus (indices), and the ontology-based recommendation rules to link resources to others. The generated website is in fact a simple single-faceted search system. Before the *SWeHG* tool the faceted photo exhibition tool *Promoottori* [57] was already developed. In *Promoottori* the facets were projected along simple ontological properties such as the *rdfs:subClassOf* property. The *SWeHG* tool introduced the more flexible rule-based projection of facets. The solutions developed for Problems 1 and 2 of Table 1.1 were later included in the *ONTODELLA* [108] facet projection and recommendation engine, which is part of the *OntoViews* [72] *FSS* framework.

The problems that surfaced along with the promising results of Chapter 3 are presented as Problems 4 - 6 in Table 1.1. The first problem that was noticed is that Semantic Web ontologies are not capable of modeling uncertainty, a problem which sometimes hinders the systems ability to provide high quality search results for the user. As a response to this we created a Bayesian method to model uncertainty in Semantic Web taxonomies. The method was first presented in my Master of Science thesis [45] in 2004, and later published in [47]. This solution contains a graph notation for representing uncertainty and conceptual overlap in Semantic Web taxonomies, and a method to compute the degrees of overlap between the concepts based on the representation. The computation can be done either directly based on the taxonomical structure, or by transforming the representation into a Bayesian network. This method has influenced the methods for modeling geo-spatial reasoning over ontology changes in time [63]. Later—as part of the work related to *fuzzy faceted semantic search*—we also created a method to weight document annotations based on their textual content. This method is an ontological extension to the *term frequency - inverse document frequency (TF-IDF)* [20] term weighting scheme.

The second problem of *FSS* was that faceted semantic search does not have the capability to rank search results according to relevance. The third problem was that con-

cepts of annotation ontologies—which are usually created for domain specialists—are not always suitable to be presented as search categories on the search GUI. As a solution to these problems we created first a fuzzy version of *FSS* which extends the crisp set theory underlying faceted search to fuzzy sets [113]. Fuzzy *FSS* is based on weighted—i.e., fuzzy—annotations. The results to searches specified by the user are not crisp but fuzzy sets of search items. The degree of membership of a search item in the result set is interpreted as the degree of relevance of the search item. Fuzzy *FSS* also includes a facility to define end-user facets separately from the annotation ontologies, and then fuzzily map these facets to concepts of annotation ontologies. This separation of search facets from annotation ontologies and the mappings between the two has been utilized in the *HealthFinland* [60, 100] portal prototype, in which the method is combined with card sorting technique [92, 76] to create the end-user facets.

Fuzzy *FSS* is a promising and a relatively simple way to extend crisp *FSS* to enable ranking and separation of end-user facets from annotation ontologies, and according to our evaluations it provides good ranking results. However, fuzzy logic has been criticized of its heuristic nature (see e.g. [98, 26]). In addition, we wanted to be able to combine evidence from multiple ranking schemes, because this has been shown to improve ranking performance [70]. As a result we started to develop the probabilistic *FSS*.

In probabilistic *FSS* a Bayesian probability model of the search system is created, and search results are ranked based on the probability of relevance of each item. Similarly to fuzzy *FSS*, also probabilistic *FSS* supports the mapping of separate end-user facets to annotation ontologies. However, in contrary to fuzzy *FSS*, the probabilistic version supports the usage of more than one ranking scheme to provide the final ranking. After the fuzzy and probabilistic *FSS* frameworks are presented, they are evaluated and compared using a real-world dataset.

As mentioned, many of the chapters in this dissertation are based on published scientific articles [49, 47, 48, 46]. The *SWeHG* tool presented in Chapter 3, which is based on [49] was originally developed in a student project, by a team of students which I was a member of. Guidance to this project was given by Eero Hyvönen and Kim Viljanen. The project was an Extreme Programming [22] style project, in which the roles were loose and in fact the project manager role was circulated among members. My main contribution during the software project was in the design and implementation of the facet projection, and recommendation rules using SWI-Prolog. Later I was part of the team that wrote the scientific paper [49]. In all of the other publications I am the main author, and they in fact present my personal work, which was helped and sparred by the co-writers. This is also true for the yet unpublished parts of the dissertation.

1.5 Organization of the Dissertation

The rest of the dissertation is divided into the following four parts:

Crisp Faceted Semantic Search This part consists of Chapters 2–4. Chapter 2 contains a brief introduction to *faceted search*, and to the main ideas of *faceted semantic search*. Chapter 3 then presents the single-faceted search tool *SWeHG* along with the facet projection, and recommendation solutions that have been used in *FSS* ever since. After these results are presented, some problems of *FSS* are raised in Chapter 4, which are then answered in later parts of the dissertation. Here, the usually used name *faceted semantic search* is prefixed with the term *crisp* to distinguish it from the *fuzzy*, and *probabilistic* extensions that are presented later in the dissertation.

Fuzzy Approach This part consists of Chapters 5–8. Chapter 5 presents some necessary background knowledge about *fuzzy sets*, and how they have been

combined with Semantic Web ontologies. Chapter 6 then presents the *fuzzy FSS* framework which is the main result of this part. Chapter 7 augments the results of Chapter 6 by providing a method for weighting document annotations based on the document's textual content. Chapter 8 summarizes the contributions and remaining problems of the *fuzzy approach*.

Probabilistic Approach The *Probabilistic Approach* part consists of Chapters 9–12. Chapter 9 consists of necessary background information about the probabilistic methods applied in the rest of this part, a short review of probabilistic search methods, and also of probabilistic methods to model uncertainty in Semantic Web ontologies. Chapter 10 presents a new method for modeling uncertainty in Semantic Web taxonomies. This method is utilized in *probabilistic FSS* which is presented in Chapter 11. An implementation of the *probabilistic FSS* framework is presented in Appendix A. Chapter 12 summarizes and discusses the contributions of the *probabilistic approach*.

Evaluation and Conclusions In the last part of the dissertation we present an empirical evaluation of the ranking methods presented earlier in the dissertation (Chapter 13). After this, in Chapter 14, the results of the dissertation are reviewed, lessons learned are elaborated, and directions for future research outlined.

Part I

CRISP FACETED SEMANTIC SEARCH

2 Introduction to Faceted Semantic Search

This chapter provides the necessary background information about faceted search that is needed in order to fully understand what faceted search is about, and how the work done in this dissertation contributes to it.

2.1 How Faceted Search Works?



Figure 2.1: The start page of the *HealthFinland* faceted health portal.

Faceted search is explained here from end-user point-of-view using a realistic ex-

The screenshot shows the HealthFinland portal interface. At the top left is the logo 'terveysuomi.fi prototyppi'. A search bar is at the top right. The main content area is titled 'HOME' and 'INDEX A-Z'. Below this, there are navigation options: 'Topic areas > Intoxicants > smoking' and 'Narrow your search'. On the left, there are several faceted search filters: 'Life event', 'family and relationships', 'Group of people', and 'Body part'. The 'Life event' filter is expanded, showing categories like 'course of life', 'ageing', 'refugeedom', 'menopause', 'youth', 'pregnancy', 'breastfeeding', 'puerperal period', 'leisure', 'travel', 'work and studies', and 'sairausloma'. The 'Group of people' filter is also expanded, showing categories like 'sex', 'age', 'role', 'occupations', 'immediate family', 'experts', 'family members', 'fetus', 'parents', 'schoolchildren', 'pensioners', 'professionals', 'risk groups', 'relatives', 'students', 'Finns', 'health service personnel', 'smokers', and 'mothers'. The 'Body part' filter is also expanded, showing categories like 'face', 'conjunctiva', 'body', 'body', 'organism', and 'reproductive system'. The main search results area is titled 'smoking' and shows a list of search results with titles and counts. The results include: 'smoking cessation (37)', 'cigarette smoke (21)', 'smoking prevention (8)', 'snuff (10)', 'tobacco consumption (7)', 'tobacco industry (15)', 'nicotine (15)', and 'passive smoking (6)'. Below the results, there is a 'Search results' section with '194 hits' and 'pages 1 2 3 4 5 6 7 8 9 10 Next'. The results are displayed in a list format with titles and brief descriptions. On the right side, there are several sections: 'information page', 'frequently asked questions', 'contact information', 'magazine article', 'organizational info', and 'test'. The 'information page' section includes links to 'Tupakkaverkko', 'Tupakasta vieroitus', and 'Nikotiiniriippuvuus ja nikotiinivieroitusoireyhtymä'. The 'frequently asked questions' section includes 'Usain kysyttyä'. The 'contact information' section includes 'Virtuaalivieroitus internetin avulla'. The 'magazine article' section includes 'Tupakan toukokuu, lopeta ja voit!', 'ENPAT: asiantuntijaverkosto ehkäisee nuorten tupakointia', and 'Tupakan tuotvastuun aika olisi jo'. The 'organizational info' section includes 'European Network on Young People and Tobacco - ENYPAT', 'EU-Web - projekti tupakan verkkovieroituksesta', and 'Pohjois-Karjalän aikuisväestön terveyskäyttäytyminen 1978-2000'. The 'test' section includes 'Testaa riskisi sairastua tyypin 2 diabetekseen'. At the bottom of the page, there is a footer with 'Find: ontodella', navigation buttons, and a 'Clear' button.

Figure 2.2: The result list page of the *HealthFinland* faceted health portal for the selection of the search category *Smoking*.

ample adapted from the *HealthFinland* health portal demonstration video⁶. Anna is a smoker who wants to have a baby. She is concerned about the health effects of smoking on the child, and wants to learn more about the subject. She goes to the *HealthFinland* health portal. The starting page is shown in Figure 2.1. In the middle of the screen the main topics of the portal are shown. These topics, in fact, constitute the main facet of the portal. On the top right there is a free-term search box, which augments the search facets, and enables normal keyword-based search of the portal's content.

Anna selects the search category *Smoking* from the topics. As a response to this

⁶<http://www.seco.tkk.fi/applications/terveysuomi/ui-presentation.html>

The screenshot shows the HealthFinland website interface. At the top, there is a search bar and navigation links for 'HOME' and 'INDEX A-Z'. The main content area is titled 'smoking + pregnancy' and displays search results. On the left, there are several faceted filters: 'Life event', 'Group of people', 'Body part', 'Type', 'Publisher', and 'Audience'. The search results are listed in the center, with each entry showing a title, a brief abstract, and a link to the full document. On the right, there are sections for 'information page', 'frequently asked questions', 'contact information', 'magazine article', 'test', and 'organizational info'. The browser's address bar shows the URL 'http://terveysuomi.fi/...' and the search results are displayed in a table format.

Figure 2.3: The result list page of the *HealthFinland* faceted health portal for the search category selection of *Smoking* and *Pregnancy*.

selection the page shown in Figure 2.2 appears. The list of documents matching the selection *Smoking* are shown in the middle of the page. For each search result, the title, a small passage from the document, and some other metadata is shown. Above the list of search results the selected topic is shown, and below it subtopics. Next to each subcategory a number is shown, which indicates the number of search results matching that selection. On the left the other facets are shown. In our case these are *Life event*, *Group of People*, *Body Part*, *Type*, *Publisher*, and *Audience*. The available search categories with a number indicating the search results matching that selection are shown under each facet title. The user can refine the search by making a selection either from the subcategories shown in the top center of the page or from one of the facets shown on the left.

The screenshot shows the HealthFinland website interface. At the top, there is a search bar and the site logo 'terveysuomi.fi prototyypä'. Below the logo, there are navigation links for 'HOME' and 'INDEX A-Z'. The main content area features a document titled 'Tupakoinnin vaikutukset raskauteen' (Effects of smoking during pregnancy) by Ingrid Antikainen, published on 20.5.2008. The document text discusses the health risks of smoking during pregnancy, including effects on the fetus and the mother. A sidebar on the right contains several sections: 'information page' with links to 'Päihteidenkäyttö ja raskaus - alkoholin vaikutuksia raskaudenlukuun ja sikiöön', 'Raskaudenaikaisia häiriöitä - alkuraskaus', and 'Synnytyksen hätätilanteita'; 'guide' with 'Mita on katkokävely?'; 'frequently asked questions' with 'Usein kysyttyä'; 'test' with 'Terveiden riskitekijät -testiä' and 'Testaa riskisi sairastua tyypin 2 diabetekseen'; 'magazine article' with 'Maailman tupakka ja terveys konferenssi 2003, The World Tobacco Or Health Conference 2003', 'Hyvän ikääntymiseen Päivät-Hämeessä', and 'Raskaudenaikainen ravitsemus'; and 'organizational info' with 'Päihderiippuvuus' and 'Päihteiden käyttöä ja mielenterveyttä koskevat kaksostutkimukset'. A publisher logo for 'Savonia ammattikorkeakoulu' is also visible.

Figure 2.4: The document page of the *HealthFinland* faceted health portal.

Anna selects *pregnancy* from the facet *Life event*, because she is interested in the effects of smoking on pregnancy. The page is shown in Figure 2.3. Now the result list contains documents that match both *smoking* and *pregnancy*. In this manner the user can browse the content according to multiple facets simultaneously. The facets on the left and the subtopics in the top center of the page are updated to match the new situation. According to the principles of faceted search it is not possible to make a selection which results in an empty result set. For this reason some of the search categories on the left are hidden and the subtopics in the center are disabled. Also the numbers next to the search categories are updated to match the new situation.

Anna selects one of the search results. Now the page shown in Figure 2.4 containing the selected document is shown. On the right, recommendations of similar documents are shown, which are in this case organized according to the *Type* of the item. Here *Type* is one of the facets of the system.

A facet often, but not always, has a hierarchical structure. In the example of Figures 2.1–2.4, both the facets *Topics* and *Life Event*, have a hierarchical structure. The selected search category *smoking* selected from the facet *Topics* is a subcategory of *intoxicants*, and supercategory of e.g. *smoking cessation*, *smoking prevention*, and *nicotine*. The other selected search category *pregnancy* is a subcategory of *family and relationships*. When a user selects a search category from a facet he/she is actually making an *OR* query using the selected search category and the transitive closure of its subcategories. When a user makes selections from multiple facets he/she is actually making an *AND* query of the individual facet queries. Thus, faceted search can be seen as an easy and intuitive way to create Boolean queries, which have been shown to be very effective, however, often not widely used, because non-expert users often have problems constructing them [43].

2.2 Set-theoretic Interpretation of Faceted Search

In this section we will present a set-theoretic interpretation of the faceted search framework. The fuzzy and probabilistic extensions to faceted search that will be presented in Parts II and III of this dissertation will build on this set theoretic interpretation.

In terms of set theory, a search category SC_j can be defined by its result set; i.e., $SC_j = \{D_i | D_i \text{ is a search item directly or indirectly annotated with } SC_j\}$. Thus, each individual annotation of a search item D_i according to a search category SC_j is interpreted as an explicit statement of direct membership of D_i in SC_j ; i.e.,

$D_i \in SC_j$. By indirect annotations we mean annotations to a subcategory of SC_j . This is consistent with the last paragraph of Section 2.1, where it was stated that a selection of a search category SC_j is interpreted as an *OR* query of SC_j and the transitive closure of its subcategories. In set-theoretic terms the subcategory relationship is defined as the subset relation between the involved search categories. Specifically, if the search category SC_k is a subcategory of SC_j , then $SC_k \subseteq SC_j$, which means that all search items belonging to SC_k belong also to SC_j . Facets are sets of search categories; i.e., sets of sets of search items. Intuitively, the search items are the individual objects of concern in faceted search, and the other objects (facets, search categories, annotations) are used just to enable the efficient finding of search items. Thus, it is natural that the search items compose the universal set of the faceted search framework.

Finally, the last paragraph of Section 2.1 stated that selections from multiple facets are interpreted as an *AND* query of the individual facet queries. In our set-theoretic interpretation this corresponds to the intersection of the result sets corresponding to the selected search categories. Thus, the search S specified by the user is defined as $S = \bigcap SC_l, \forall l \in 1, \dots, m$, such that SC_1, \dots, SC_m are the search categories selected by the user, and each SC_l belongs to a different facet. Notice, that we do not have to deal with the case of the user making multiple selections from a single facet, because by convention the user can only select one search category from one facet at any given time.

2.3 Facets and Metadata

Faceted search is based on metadata attached to the searched items. This metadata is organized according to facets, that is, composed of orthogonal sets of categories as exemplified above. This organization of metadata is called *faceted classification* [86] and it was originally created to classify library collections [89]. In *faceted classifica-*

tion the searched items are classified according to each of the orthogonal facets. In addition, each item might belong to more than one category of each facet, however, some facets might allow only one category per item [43]. Thus, *faceted classification* differs significantly from the widely used Dewey’s library classification [31], where all items are classified according to one monolithic taxonomy, and each item belongs to exactly one category of this taxonomy.

One of the challenges of faceted search is how to create these facets, and how to assign the metadata to the items accordingly. Often, the facets are created using one, or a combination, of the following approaches:

Using an existing vocabulary or thesaurus If a suitable vocabulary exists, it can be used as a facet as such, or by selecting a part of it.

Manual creation If a suitable vocabulary is not available, a facet vocabulary can be created by hand according to the specific need. In many cases, card sorting or similar method is used in the creation of these facets [100].

Analyzing the existing metadata and content of the items to classify Some facets—e.g., *Creation date*—can be created automatically based on existing metadata of the items. This applies to facets that are based on metadata that is typically attached to items in a standard format using standard metadata fields, such as fields specified by the *Dublin Core* [1] metadata scheme.

After the facets themselves are created the search items are described—i.e., indexed, annotated—according to these facets. This is depicted in Figure 2.5. Typically, some of this indexing can be done automatically, e.g., indexing according to the *Creation date* facet above, and at least some of it has to be done manually, e.g., indexing according to a manually created vocabulary above, which might be an arduous and resource consuming process [100, 43].

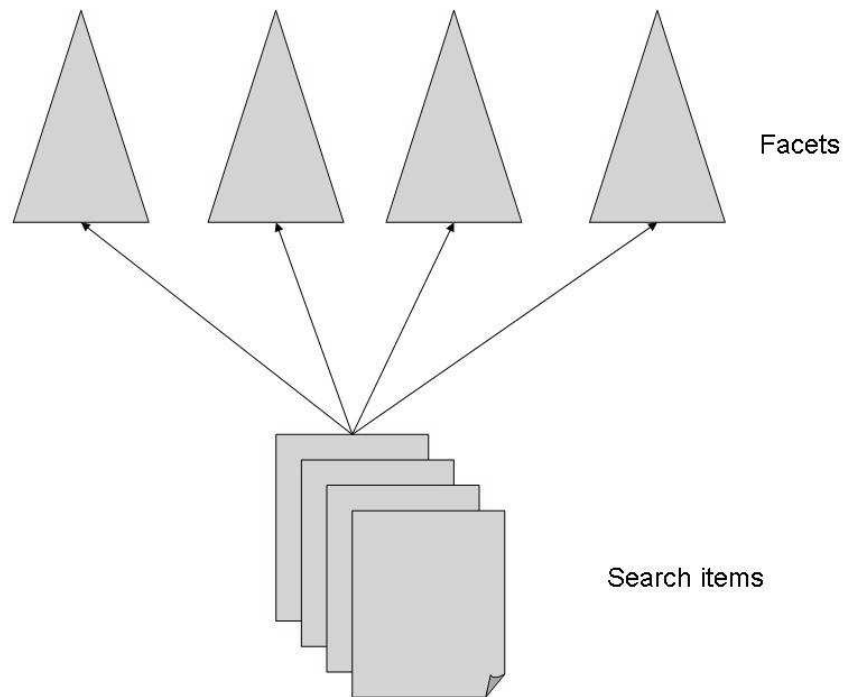


Figure 2.5: Depiction of faceted classification. Search items are classified according to each of the facets.

2.4 The Promise of Faceted Semantic Search

According to the vision of the Semantic Web, current Web content will be described, i.e., annotated using ontological concepts. Thus, rich machine readable metadata will be available for at least a large part of Web content, which could be utilized to construct facets algorithmically from a set of underlying ontologies that are used as the basis for annotating search items. Furthermore, the mapping of search items onto search facets could be defined using logic rules. This facilitates more intelligent *semantic* search of indirectly related items. Another benefit is that the logic layer of rules makes it possible to use the same search engine for content annotated using different annotation schemes. Ontologies and logic also facilitates *semantic browsing*,

i.e., linking of search items in a meaningful way to other content not necessarily present in the search set.

2.5 Content Creation for Faceted Semantic Search

This dissertation focuses on the utilization of existing ontologies and databases of documents annotated according to these ontologies. This kind of data already exists in rather large quantities, and faceted search systems are being built on such knowledge bases both in research and commercial settings. However, in order to complete the picture, a short discussion of content creation methods is presented in this section. This discussion is largely based on [101] and [62].

2.5.1 Ontology Creation and Reuse

Typically a faceted semantic search solution is implemented as a part of a Web portal, where content is gathered from multiple source organizations [101, 51, 52]. The used ontologies should describe all the relevant concepts in the application domain of the portal. In addition, there should be a relative consensus among the participating organizations about these concepts and the relations between them. For these reasons pre-existing, established ontologies and structured vocabularies are often used as the basis of ontologies. Reusing existing vocabularies offers also several other advantages: it eases semantic interoperability with other applications, saves time and money by avoiding unnecessary ontology engineering work, and helps to ensure broad coverage of the subject area as established ontologies typically have been used and developed for a long time [101].

Often, standard vocabularies are expressed as thesauri. In Finland, for example, there are numerous thesauri conforming to the ISO [2788] standard [15], such as

YSA⁷ which is the general finnish thesaurus, MASA [71] which is a thesaurus for the subject domain of museums, and Agriforest⁸ which is an agriculture and forest thesaurus.

In the National Semantic Web Ontology project (FinnONTO 2003-2007) [59] a method was developed for transforming thesauri into ontologies [62]. The method is not purely syntactic, because thesauri, typically, are semantically loose and contain relations that require implicit background knowledge by the user, whereas the idea of ontologies is to define the meaning of concepts explicitly and accurately enough for the machine to use. Instead, the method developed in *FinnOnto* was based on criteria from DOLCE [37], and the transformation is done by refining and enriching the semantic structures of the transformed thesaurus. According to the method the thesaurus is first syntactically transformed into RDF/SKOS [10]. Then the following enrichments to the thesaurus structure are done using the Protegé ontology editor⁹:

Completing the hierarchies The thesaurus hierarchies typically are incomplete and arrange the terms into separate smaller subhierarchies. For this reason the hierarchies have to be completed. A central structuring principle in constructing the hierarchies is to avoid multiple inheritance across major upper ontology categories [62].

Removing ambiguity of the BT relation The hierarchical Broader term (BT) relation used in thesauri may mean either subclass-of relation, part-of relation, or instance-of relation. The BT relation is transformed into subclass-of and part-of relations according to the current semantics, but instance-of relations are not used.

Assuring transitivity of concept hierarchies The BT hierarchies are not always transitive in the ontological sense, i.e., an instance of *A* is not necessarily

⁷<http://vesa.lib.helsinki.fi/>

⁸<http://www-db.helsinki.fi/triphome/agri/agrisanasto/Welcomeng.html>

⁹<http://protege.stanford.edu/>

instance of B even if B is a broader term of A . Thus, the hierarchies are cleaned from non-transitive hierarchical structures.

Removing ambiguity of concept meanings Terms in thesauri often contain multiple meanings. E.g. the term *child* can mean a family relation or a period of human life. Ambiguous terms are split to multiple concepts according to the different meanings.

The *FinnOnto* project also developed a semi-automatic method for aligning ontologies. In this method, ontology classes are first automatically aligned based on term labels using equivalence, then imported into Protegé for further editing and checking by a human expert. In this scenario, the ontologies are usually not aligned pairwise, but instead domain ontologies are aligned to the Finnish General Upper Ontology (YSO)¹⁰, which is an ontologized version of the general finnish thesaurus YSA. YSO acts as a glue or mediator between the domain ontologies.

An alternative to the above described thesaurus based ontology creation process would be to use algorithms for ontology learning. The main idea of these algorithms is to use available—typically textual—information sources such as webpages, dictionaries, knowledge bases out of which concepts and concept-relations are extracted to form an ontology [40]. Usually these ontology learning algorithms aim at partial automatization of the ontology creation process. The ontology learning methods can be grouped as follows:

Linguistic Analysis These methods are based on linguistic techniques such as linguistic patterns, pattern-based extraction, semantic relativeness measures etc. One example would be using common linguistic patterns in text, such as, *a birch is a tree*, to infer subclass-of relations, such as *birch subclass-of tree*. For more information see e.g. [17, 19, 42].

¹⁰<http://www.seco.tkk.fi/ontologies/ys/>

Statistical Analysis Statistical methods use statistical measures to help the ontology engineer to detect new concepts or relations among concepts. Statistical techniques are often applied together with other techniques such as natural language processing. For example, Faatz and Steinmetz present a statistical method that uses the Web as a corpus, to suggest new concepts to an existing ontology [33]. Agirre et al. present a method [14] that enriches the relations among concepts in an existing ontology using both the Web and WordNet [34].

Machine learning These methods base ontology learning on machine learning algorithms, for detecting new concepts or relations among them. Machine learning techniques are usually applied together with natural language processing techniques. For example, Cimino and Staab [28] have developed a method that learns concept hierarchies based on WordNet, the Web, and other text collections.

An empirical comparison of the thesaurus based and the ontology learning approaches has not been performed. For this reason we do not know the difference between the approaches in terms of manual work needed, as well as the quality of the resulting ontologies. However, in situations where a thesauri is not available ontology learning algorithms offer a viable solution.

2.5.2 Annotation of Content

To enable use of heterogeneous content from multiple sources a faceted semantic search portal needs a metadata or annotation schema in addition to the common ontologies discussed above. This annotation schema defines what metadata the portal needs for each published document, and how this metadata should be represented. Typically this annotation schema is based on the Dublin Core metadata

standard [1] to maximize reusability of content.

Metadata creation requires suitable tools that aim to assist the annotators in the process. In the *FinnONTO* project a number of solutions to this problem have been developed:

Extending the content producer’s CMS Often content producers of a faceted semantic search portal have a content management system (CMS) to produce and publish Web documents on their own site. Adding support for metadata production using the portal’s metadata schema and ontologies makes it easy for content authors to produce metadata and keep it synchronized with updates in the content. In such a setting many metadata fields such as publication and modification dates can be automatically assigned values. The *FinnOnto* project has developed a National Ontology Library Service ONKI, which includes an ONKI Selector Widget [61], which enables the easy extension of a CMS to support the needed annotation schema features. The ONKI Widget is an AJAX-enabled component that can easily be integrated into any HTML form and it allows the selection of concepts from any vocabulary available on the ONKI server. The CMS can then be configured to publish this semantic metadata either into individual HTML pages or as RDF data that is retrievable by the faceted semantic search portal [101].

SAHA To create metadata for content producers that do not use a CMS a browser-based metadata editor SAHA has been developed [106]. SAHA enables the manual distributed creation of metadata according to an annotation schema using shared domain ontologies.

Semi-automatic annotation The POKA tool has been developed in the *FinnONTO* project to enable the semi-automatic annotation of text documents [107]. The tool finds candidate annotations from a text document by matching labels of ontology concepts to words in the document. The human annotator then

chooses the best ones to be used as annotations. Naturally, the human annotator can also add annotations that are not suggested by POKA.

Also other approaches to automatic annotation exist. For example, Mukherjee et al. [82] present an approach for automatic annotation of HTML documents with semantic labels. In the approach, HTML documents are partitioned into semantic structures, and it incorporates the use of ontologies and lexical databases such as WordNet. As another example, Yang [112] presents an approach to automatically annotate documents that employs techniques of ontology and linguistics. The method is part of a larger effort to develop semantic portals.

2.5.3 Attachment of Uncertain Information to Ontologies and Annotations

Parts II and III will present methods to represent and reason about uncertainty in faceted semantic search systems. This uncertain information is mostly intended to be attached automatically to the ontologies and annotations. This methods will be discussed in detail in the relevant parts of this dissertation, however, here is a short summary about the automatic methods to attach uncertain information to a faceted semantic search system that are described in this dissertation:

1. Chapter 7 will present an ontological extension to the TF-IDF term weighting scheme for automatically weighting annotations with a real number in the range $[0, 1]$.
2. Chapter 10 assigns probabilities to a geographical taxonomy based on geographical knowledge which could be extracted automatically from a geographical knowledge base.
3. Chapter 10 also presents a method to assign probabilities to any concept taxonomy based on a document set annotated according to the concepts of

this taxonomy.

4. Chapter 13 presents a technique to weight annotations based on the total number of individual annotations that a document has.
5. Chapter 13 also automatically assigns probabilities to ontological relations based on the semantics of the relation.

2.5.4 Evolution of Content

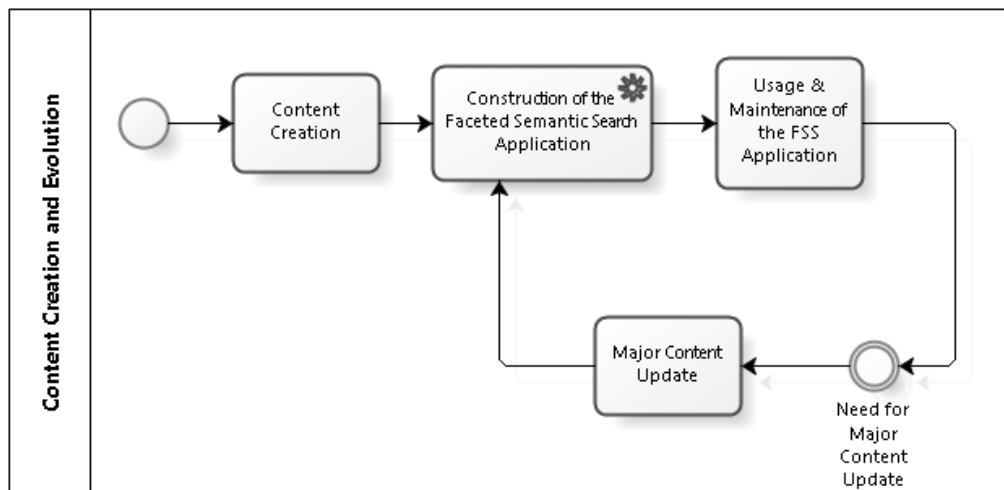


Figure 2.6: Content Creation and Evolution

Figure 2.6 outlines the typical major phases in a faceted search application from the point-of-view of content creation and evolution:

1. **Content Creation.** The first phase is the creation of content, and it corresponds to the ontology creation and the annotation of content activities described in Sections 2.5.1-2.5.3 above.
2. **Construction of the Faceted Semantic Search Application.** The second phase is the algorithmic construction of the faceted semantic search

application based on the created content; i.e., the projection of facets and possibly recommendation rules based on the underlying semantic knowledge base. Chapter 3 describes the construction of a single faceted semantic search application, Chapter 6 presents the construction of a fuzzy faceted semantic search application, and Chapter 11 presents the construction of a probabilistic faceted search application. In principle, this phase could be executed on demand when the end-user performs a search in the FSS application, however, typically essential parts of the application functionality are precomputed.

3. **Usage and Maintenance of the FSS Application.** After the faceted semantic search application is constructed it is used by the end-users. This phase is the purpose for the development of the application in the first place. This phase will typically include maintenance activities, such as the addition of new documents, the updating of the metadata of existing documents etc. In these situations each new or updated document will be dynamically indexed according to the facets.
4. **Major Content Update.** At some point during the usage of the application a need for a major content update may arise. This may be due e.g. to evolution in the underlying thesauruses, the incorporation of a new content provider to the portal. In this phase content is updated using the methods and tools described in Sections 2.5.1-2.5.3 above. After content is updated the faceted semantic search application is constructed again according to Step 2 above.

This dissertation focuses on the second phase of the process described above. The proper handling of the other phases is not less important, however, they are scoped out of this dissertation because each of these phases deserves a dissertation in its own right.

The next chapter presents a tool for publishing Semantic Web content as a website

that is organized according to facets, and recommendation links, that are generated algorithmically from the ontologies used to annotate this content. The solutions developed in this chapter have later been used as facet projection and recommendation link creation methods of the *OntoViews* faceted semantic search framework. Notice, however, that *SWeHG* in itself not a fully featured faceted semantic search tool. Namely, *SWeHG* lacks the possibility to make selections from multiple facets simultaneously. For this reason sites created using *SWeHG* can be called *single-faceted semantic search* applications, to distinguish from the usual *FSS* portals that are, in fact, *multi-faceted semantic search* applications.

3 Designing and Creating a Website Based on RDF Content

This chapter is largely based on the article:

Eero Hyvönen, Markus Holi, and Kim Viljanen. 2004. Designing and Creating a Web Site Based on RDF Content[49]. In: WWW2004 Workshop, Application Design, Development and Implementation Issues.

3.1 Two Views of the Semantic Web

The notion of the Semantic Web [23, 35] has two interpretations. From the machine's viewpoint, the Semantic Web manifests itself as a distributed source of interpretable metadata concerning resources, such as webpages, documents, photos, and real world objects. The metadata descriptions are given in terms of ontologies using frameworks and languages such as RDF(S) and OWL. From the human's viewpoint, the Semantic Web looks like the current Web; i.e., it is a repository of HTML pages, but empowered with more useful semantics-based links, search engines, and intelligent Web services.

A central question in the development of Semantic Web applications is how the content represented for the machine can be transformed for the human to view; i.e., how machine interpretable RDF(S) or OWL content proliferating the Web can be rendered to the human end-user as a searchable and browsable HTML website or space. In this chapter we present a genuine approach and tool named *Semantic Web HTML Generator (SWeHG)* [58] to address this problem (See Figure 3.1). The idea is to specify the structure and the layout of an HTML website in terms of a set of HTML templates using a tag language. The templates can be used by a Web

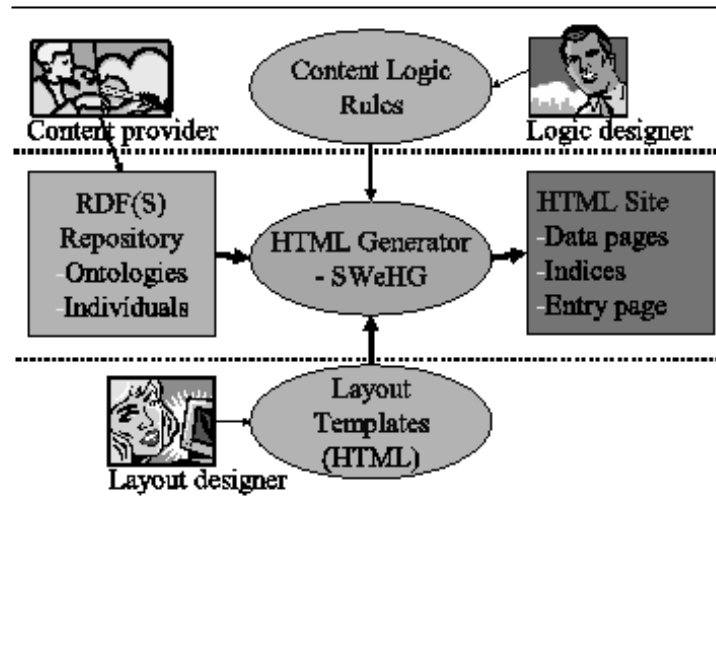


Figure 3.1: Rendering RDF(S) content as an HTML website.

layout designer who does not know the details of the underlying RDF(S) content or Semantic Web technologies. The semantics of the tags—i.e., the machine’s view on the RDF level—is specified by a Semantic Web programmer in terms of logic predicates. A benefit of separating the HTML and RDF levels is that ontological details and variance can be hidden from the HTML designer. By modifying the semantics of the tag, content represented using different ontological structures can be mapped on the same HTML tags that the HTML designer is capable of using. The tag definitions can be re-used directly in applications based on similar ontologies and annotation schemes. The templates provide a declarative description of the website structure, indices, and linkage. By modifying the templates alone in HTML, the same RDF(S) content can more easily be rendered in different ways in different applications to human end-users. In the following, we first discuss two examples of semantically indexed and linked HTML websites generated by SWeHG. The layout specifications with the corresponding tag definitions needed for the RDF

to HTML transformation are then discussed. After this, the transformation process and its implementation are presented. In conclusion, experiences of our research and experimentation are summarized, related work is described, and directions for further research are outlined.

3.2 Example Applications

As an example of using SWeHG, the virtual exhibition of a photo archive in the Helsinki University Museum was generated. The archive contained 629 photographs about the promotion ceremonies of the University of Helsinki. The content of the archive was transformed into RDF(S) format in an other application project [53] and was used as it is by *SWeHG*. The domain knowledge consists of six ontologies with 329 promotion-related concept classes, such as *Person* and *Building*, 125 properties, and 2890 instances, such as *Linus Torvalds* and the *Entrance of Cathedral of Helsinki*. In the photo annotation schema, the subject of a photograph is represented by a collection of ontology classes and individuals that appear on the image. For example, if Linus Torvalds appears in a photo on a particular street, then the photo record is related directly with the corresponding person and street resources with a property corresponding to `dc:subject`.

However, the relation between photos and subjects can be indirect, as well, involving traversal through several RDF arcs in the underlying knowledge base. For example, Linus Torvalds is present in a photograph as a Honorary Doctor. Then only an instance of such a role is associated with the image. The person instance is not directly linked with the image, but indirectly through the role instance. SWeHG predicate definition facility is very handy in hiding such annotation schema specific details from the HTML designer: the persons can be associated with images either directly or indirectly through roles. The criterion for association can be defined freely and conveniently by a declarative predicate.

Using SWeHG to publish the archive provides the end-users with two services. First, the photos can be found along the different orthogonal facets based on the ontologies. Second, the photos can be browsed by using the links created between semantically related photos. The links are grouped based on the semantics of the link. For example, there is a link group that points to other photos taken of the same person.

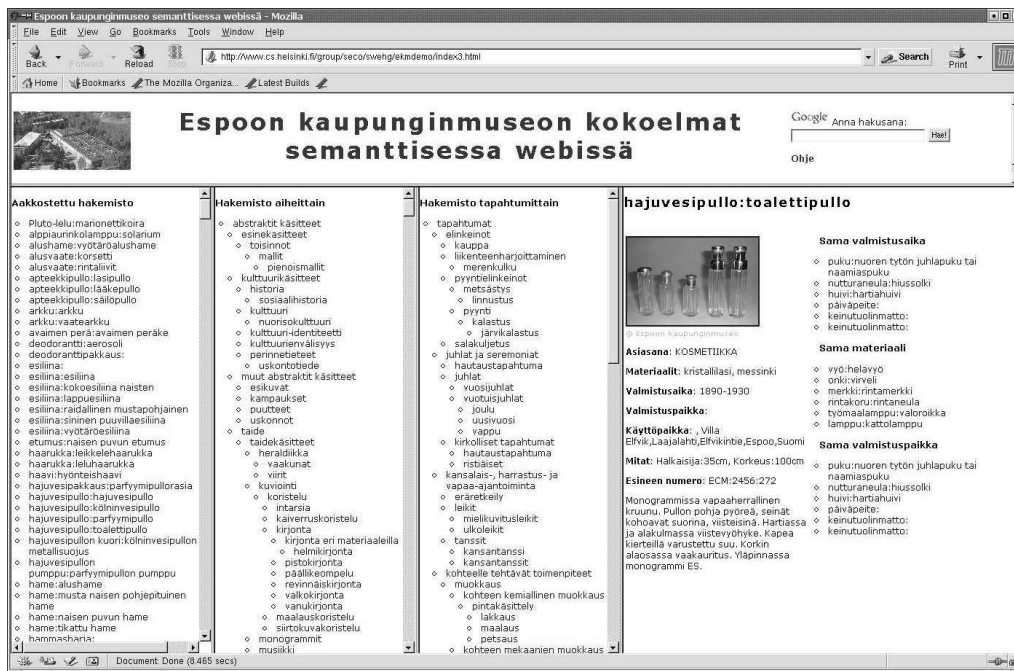


Figure 3.2: A photo exhibition generated with SWeHG.

Figure 3.2 presents the home page of another example application of SWeHG, i.e., the exhibition *Espoo City Museum on the Semantic Web*. Seven RDF(S) ontologies are used with some 10,000 classes and individuals and the metadata is described in terms of 38 properties. The RDF(S) repositories were originally created for the semantic portal MuseumFinland [52]. In this work, we could re-use the semantic recommendation predicates and the inference rule base developed for the original system, and the exhibition could be generated in a day or two.

In the RDF(S) repository, each ontological property of the collection objects in the exhibition, such as *material* is associated with a domain ontology of its own.

For example, artifact, material, and technique ontologies have been defined based on the Finnish MASA Thesaurus [71] of keywords used in several museums for indexing data. The ontology MAO [55] created based on MASA contains some 6600 classes organized in a taxonomy. There is also a location ontology that defines geographical concepts such as *country* and *town*. Their instances are individual areas and places. The places are related with each other by a part-of meronymy. In the same way, an agent ontology defines concepts such as *person* and *company*, whose instances are active individuals. There is also an ontology for time periods. Still another ontology of *activities and processes* contains a taxonomy of concepts such as *wedding* and *fishing*. It is used to provide the end-user with an event-based view to cultural artifacts by associating them with corresponding events through annotations and logical rules. Each object's metadata and annotations are given in an RDF instance, that points to different classes and instances of the ontologies by the respective URIs through RDF properties. Some of the properties in an RDF card have literal values, and some point to resources by using URIs.

The created HTML site consists of some 1200 resource webpages (RPage) describing objects in the museum's collection database, pages indexing the contents along different classifications, and a short user's guide. On the left in Figure 3.2, three frames containing indices for the underlying content are seen. The alphabetical index (*Aakkostettu hakemisto*) contains links to the RPages in alphabetical order. By selecting a link, the corresponding RPage is shown on the right. In Figure 3.2, the user has selected a link to an RPage depicting perfume bottles. Before making a selection, the user's guide was shown in the same frame. The classified index (*Hakemisto aiheittain*) is based on the RDFS taxonomy of the underlying cultural MAO ontology [55] that was used when creating the collection metadata. When selecting a concept, the rightmost frame shows links to its subconcepts together with links to RPages whose objects are directly related to the concept. By selecting a subconcept link there, the taxonomy can be browsed further downward; by selecting a link to an RPage, the corresponding collection object with its metadata can be

viewed in the frame. The third index *Hakemisto tapahtumittain* classifies the collection objects by associating them with the different events, processes or activities in which the objects are used or otherwise related to.

By using the indices, the user can find collection objects of interest. An alternative way is to use a conventional search engine. In the upper right corner of Figure 3.2 a form for using Google to search for the pages in the repository is seen. The hit list will be shown in the rightmost frame. After finding an PPage of interest, the collection can be browsed by using the semantic links generated between related collection items. For example, in Figure 3.2 links to objects manufactured at the same location, objects of similar material etc. can be clicked. The semantic links are generated based on the underlying ontologies, metadata, and logical recommendation rules.

The museum can publish the content by just copying the pages into a public HTML directory. This is of practical importance, since museums do not necessarily have competent IT personnel, servers, and resources to create and maintain semantic portals of their own. To sum up, the output of SWeHG is a semantically linked space of HTML pages of the following kind: 1) Resource pages (RPage) depict selected resources with their metadata. 2) Index pages (IPage) classify RPages along conceptual hierarchical classifications; i.e., facets or views [86]. By using IPages, RPages can be found along different facets. 3) A home page (HPage) defines the entrance page to the HTML repository.

3.3 Specifying the Transformation

Figure 3.3 depicts the RDF to HTML transformation. The RDF graph is on the left. Each $..R_n$ corresponds to a resource corresponding to a data entry in the RDF repository. In our example, the data entries are collection objects with their

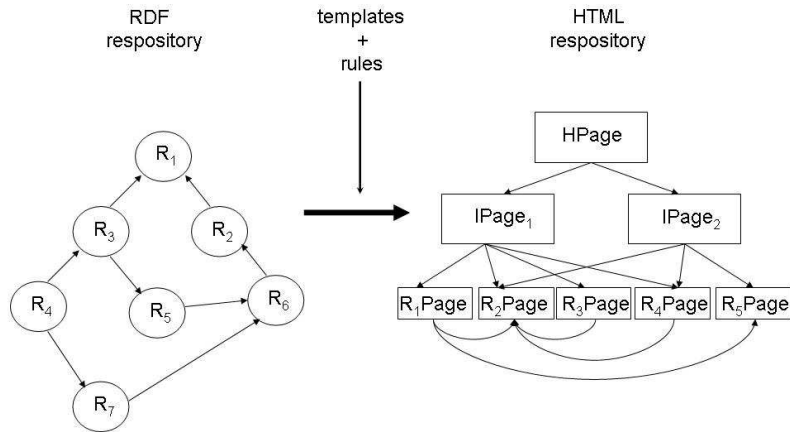


Figure 3.3: Transforming an RDF repository into HTML pages.

metadata. On the right, the HPage has links to various IPages classifying the underlying RPages that are related with each other by semantic links.

The transformation is based on descriptions on two levels: 1) The layout of the HTML pages is described on the HTML level by templates using custom tags. 2) The semantics of the tags is defined on the RDF level in terms of logical rules based on the input RDF(S) content. The idea is that an HTML designer can design the layout of the page repository to be generated by using tags without knowing details of the underlying RDF structures, RDFS ontologies, and Prolog programming. RDF(S) related knowledge as well as programming capability in Prolog is needed only for the system programmer when defining the tags. The same tag definitions can be re-used in applications conforming to similar ontological schemes.

SWeHG provides the HTML designer with three major tags: `getProperty`, `getLinks`, and `getView`. The tag

```
<getProperty name=.. >
```

is used for rendering a label related to the resource underlying an RPage. For example, the metadata property values of the bottles and the photo in Figure 3.2 are rendered in this way. The relation can be specified by the system programmer on the RDF level freely by a binary logical predicate. The tag `<getLinks>` is used for rendering links between RPages. For example, the tag

```
<swehg:getLinks
  name="SameLocation"
  listType="ul"
  listStyle="text-size: 10;"/>
```

could expand into the following HTML code linking photographs taken at the same location:

```
<ul style="text-size: 10;">
  <li><a href="entry.Mediocard_00071.html">
    View from Eiffel-tower</a></li>
  <li><a href="entry.Mediocard_00143.html">
    Cafe Parisienne</a></li>
  ...
</ul>
```

On the RDF level, the criterion *SameLocation* for the linkage could be defined by the predicate below. It associates the attribute *SameLocation* with the HTML link label *Same Place* and the predicate *photosWithSameLocation* defining the link relation.

```
swehg_relation_rule( 'SameLocation',
                    'Same Place',
                    photosWithSameLocation).
```

```
photosWithSameLocation(Context, Target) :-
    photo(Context), photo(Target),
    rdf(Context, _:place, Location),
    rdf(Target, _:place, Location),
    not(Context == Target).
```

The tag `<getView>` renders into a hierarchical index-like facet of category resources used in IPages. Each category is associated with a set of subcategories and additional individuals of the categories. A facet is defined by specifying 1) the root resource selector, 2) a binary subcategory relation predicate, and 3) a binary relation predicate that maps the hierarchy categories with the individuals used as leaves in the facet. For example, the tag

```
<swehg:getView
    roots="buildings"
    branches="subclass"
    leaves="photoOf"
    listType="ul" />
```

expands recursively into a hierarchical unordered tree (ul), where the leaves are links to photo record resources related to different building categories. The predicate definitions for the attribute values can be, e.g., the following:

```
buildings(URI) :- rdf(URI, rdf:type, 'http://some.org#building').
```

```
subclass(SubCategory, SuperCategory) :-
    rdf(SubCategory, rdfs:subClassOf, SuperCategory).
```

```
photoOf(Class, Record) :-
    rdf(Instance, rdf:type, Class),
    rdf(Record, dc:subject, Instance).
```

Here the predicate *buildings* selects the class *building* as the facet root, and the hierarchy is expanded along the *rdfs:subClassOf* property. The *photoOf* predicate relates each building type of this tree with a set of photo record resources which are used as the leaf categories. These are rendered as HTML links to the corresponding RPages. The tag definitions could also be much more complex than this, depending on the structure of the RDF(S) repository, and the desired output. The facet expansion into HTML can be controlled with the help of additional tag attributes for, e.g., ordering the categories.

The following is an example of a complete RPage template. It could be used for rendering the images using the HTML *img*-tag and links to related RPages:

```
<swehg:template selector="photo">
<html>
  <body>
    <h2><swehg:getProperty
      name="Title_Of_Photo"/></h2>
    <p>" /></p>
    <h3>Photos from the same place:</h3>
    <swehg:getLinks
```

```

        predicate="sameLocation"
        listType="ul"/>
    </body>
</html>
</swehg:template>

```

The tag attribute selector in the tag `<swehg:template>` tells the criterion for selecting context resources from the RDF repository. Each context resource will have an RPage of their own on the HTML level. The attribute value, here `photo`, is the name of a unary Prolog predicate called selector that should evaluate true for context resource URIs. An example of a complete IPage template is given below using the view—i.e., facet—definitions above:

```

<swehg:template>
<html>
  <body>
    <h1>Building index</h1>
    <swehg:getView
      roots="buildings"
      branches="subclass"
      leaves="photoOf"
      orderby="order_alphabetically"
      listType="ul"/>
  </body>
</html>
</swehg:template>

```

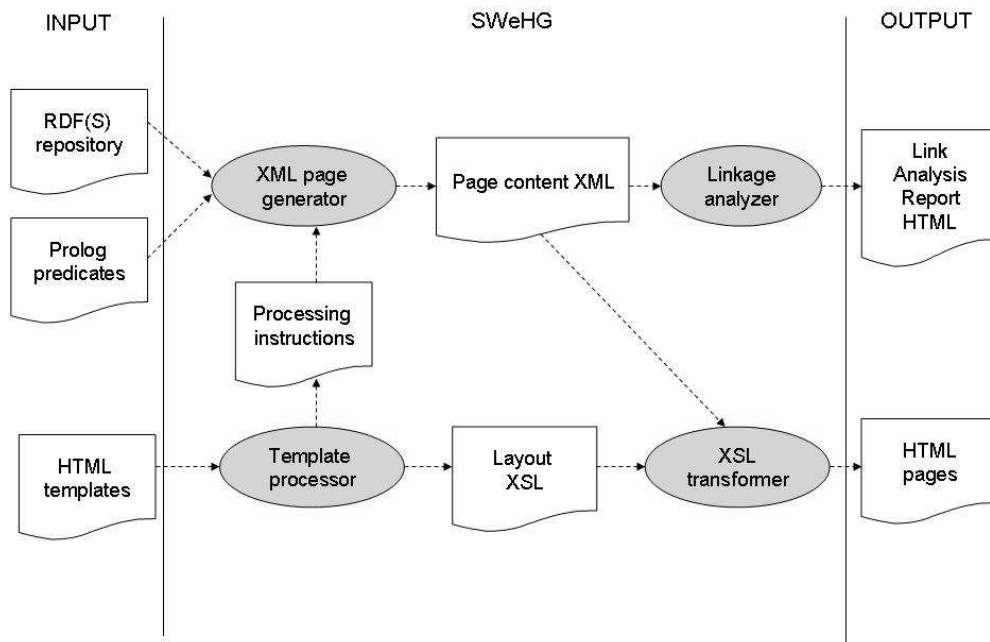
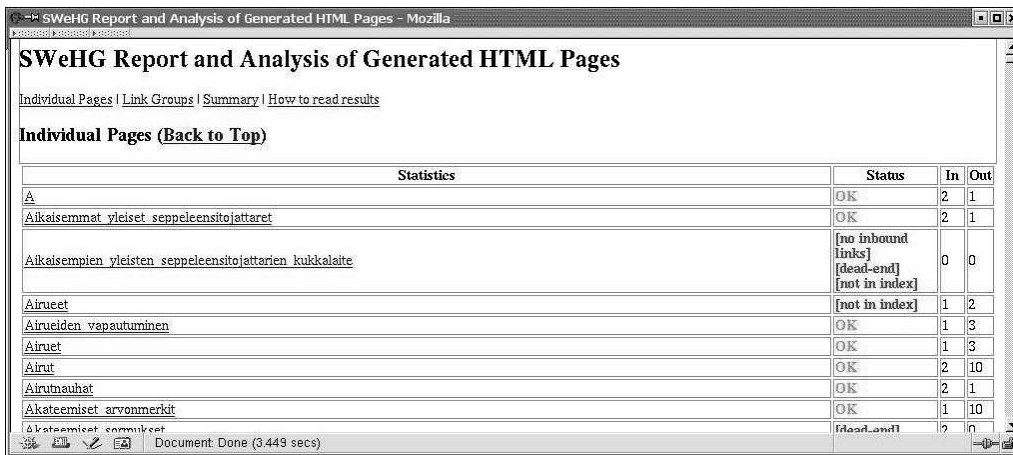


Figure 3.4: Internal architecture of SWeHG.

3.4 Website Generation

The process for transforming an RDF(S) repository into HTML pages is defined by Algorithms 1 and 2. The input of the procedure is a set of HTML templates, and an RDF(S) repository. The output is an HTML page repository conforming to the templates. The transformation is based on a set of logical rules for selectors, properties, links, and facets, which are called views in the templates tag language.

The pages are generated using the HTML templates one after another. If a template is associated with a selector, then it is expanded into a set of RPages corresponding to the selected context resources, else it is expanded once without a reference to a context resource. In the latter case, the HPage and IPages are created. When generating an HTML page, the tags are expanded into HTML in the ways described



Statistics	Status	In	Out
A	OK	2	1
Aikaisimmat yleiset seppeleensitojattaret	OK	2	1
Aikaisempien yleisten seppeleensitojattarien kukkalaite	[no inbound links] [dead-end] [not in index]	0	0
Airuudet	[not in index]	1	2
Airuuden vapautuminen	OK	1	3
Airut	OK	1	3
Airut	OK	2	10
Airumuutokset	OK	2	1
Akateemiset arvonmerkit	OK	1	10
Akateemiset tutkimukset	OK	2	1

Figure 3.5: An analysis page created by SWeHG.

in the previous section.

```

1 Algorithm:RDF2HTML
Data: Templates T, RDF(S) repository R
2 HTMLPageRepository H = empty;
3 foreach Template t in T do
4   if t has a selector rule S then
5     foreach RDF Resource r in R do
6       if S(r) == true then
7         h = createHTMLpage(r, t);
8         add h to H;
9       end
10    end
11  else
12    h = createHTMLpage(T);
13    add h to H;
14  end
15 end

```

Algorithm 1: Main procedure for the RDF to HTML transformation

Figure 3.4 depicts the architecture of our implementation. The main program is a Perl script which first builds an XSLT 9 template out of the HTML templates using the module *Template processor*. This module also writes out a set of *Processing instructions* into a separate Prolog source code file. These instructions link template tags with the Prolog predicates used in them as attribute values. The module *XML page generator* is a Prolog program that applies the predicates used in the HTML tags with respect to the RDF repository according to the Processing instructions.

```

1 Algorithm:createHTMLpage
   Data: Template t, Context Resource r
   Result: HTML page H
2 String H = t;
3 foreach Tag in H do
4   |   h = executeRule(Tag.rulename, r);
5   |   replace Tag in H with h;
6 end

```

Algorithm 2: Algorithm createHTMLpage for rendering an HTML template. Tag.rulename returns the name of the rule, e.g., *getProperty*. An analysis page created by SWeHG.

The result is a set of XML files describing the page contents. These XML files are then transformed using Apache Xalan10 and with the help of the XSLT templates generated earlier into the final HTML pages.

The intermediate XML files in Figure 3.4 are also used as a basis for the *Linkage analyzer* module that tries to identify the following potential problems: Self loops (a link that points to the page itself), Bad links (link pointing to a non-existing page), Dead ends (RPages with no outbound links), No way in (an RPage with no inbound links from any RPages or IPages), Not in index (an RPage with no inbound links from any IPage), and Unused rules (rules that are newer used when generating the HTML repository). The analysis results are represented as HTML pages. This helps the designer in debugging the specifications. Figure 3.5 depicts a portion of the result from the analyzer.

On this page the number of in-coming and out-going links can be seen for each RPage together with a status explanation. The analyzer has found out that the page with label *Aikaisempien yleisten ...* is not connected with any other page or index. Furthermore, the page *Airueet* has one incoming and two outgoing links but was not included in any index. This kind of connectivity information is vital when debugging the logical rules that produce the HTML pages.

3.5 Related Work

At the time of writing of the original article, logic and dynamic link creation on the Semantic Web had been discussed, e.g., in [21, 87]. Our approach is different in its use of HTML templates and Prolog for describing the static HTML output. In the *RDF Twig* tool [109] the RDF to HTML transformation is based on XSLT. A problem here is that an RDF graph can be serialized in many ways in XML. Different applications may produce different XML serializations of the same RDF graph, and thus a number of XSLT templates would have to be written for a single graph. In our approach only actual changes in the graph structures are relevant, because in SWI-Prolog, by which we define the logical rules, the RDF graph is processed purely as triplets. In Spectacle [99] the RDF to HTML transformation is based on APIs. Then the user must write programs that use the API, and also an application server is needed. In contrast, our approach is based on tags, and is declarative, which—we argue—makes the creation of sites technically less demanding in comparison to Spectacle. In addition, the result of a SWeHG transformation is a set of static pages whose linkage structure is inferred by logical linking predicates. The site then is not dependent on an application server, but can be served from any Web server capable of serving static HTML pages.

In the years that have passed since writing the article in 2004, the bulk of work related to publishing and utilizing of semantic content has been in the field of semantic portals and dynamic systems. Typical such systems for publishing semantic content are faceted semantic search portals which have been reviewed in [100].

3.6 Conclusions

Our initial experiences indicated that the presented RDF to HTML transformation method was feasible because HTML templates can be created fairly easily, and they can be adapted to different RDF repositories. Moreover, changes in ontology versions do not affect the usage of the templates on the HTML level in any way. The idea of using logic and Prolog for defining the semantics of the tags proved to be powerful. Complicated semantic link relations and facets can be defined and modified easily thanks to the declarative nature of logic programming. By using generic rules it is possible, in principle, to create tag definitions that will apply to any RDF repository.

In retrospect, the most durable result of this chapter's work was the use of logical rules both to algorithmically create the facets based on the annotation ontologies, and to attach the search items to these facets. This is in contrast to traditional faceted search systems, such as [86, 43], where the process of creation of facets and then indexing of search items according to these facets is often manual. The main benefit—in addition to the resource savings—is that arbitrary mappings between search categories and data resources can be flexibly defined. The system infers the mapping between facets and resources which gives it an *intelligent* flavor. Furthermore, the HTML pages are linked semantically with each other according to the ontologies, metadata, and rule base used. To the end-user, the underlying hidden associations between collection objects is a most interesting aspect of cultural collections. The nature of the associations can be explained to the user by the labels of the links.

Also the use of tag definitions to specify the layout of the HTML pages seemed a viable solution, because the tags are not application specific, and can be used also in different applications that use the same RDF(S) content. For example, we could use the linkage rules, the selector rules, and the rules generating the facets of the

indices of the *Espoo City Museum on the Semantic Web* developed originally for a semantic portal [50].

However, the tag language of SWeHG was and still is limited, and it can not be extended easily. Also, the set of different HTML outputs that the tags produce is limited. The output varies from simple strings to lists of links. In addition, SWeHG does not offer sufficient tools for testing the tag definitions before the actual transformation. A preview or debug function would be useful, because when the RDF(S) database is large, then the transformation process is long.

SWeHG generates static pages in a batch process before publishing them on the Web. This approach has the following benefits when compared with dynamic semantic portals: First, the page repository can be published easily by just copying it into a public HTML directory. SWeHG can be adapted to different contents conforming to different ontologies. Second, the publication process is independent from semantic portal providers. No special server software is needed. Third, the pages need no special maintenance. The static pages are indexed and searched for by general search engines. Fourth, the pages can be viewed efficiently. Fifth, data security problems are minimal. And sixth, the properties of the resulting HTML page set can be analyzed efficiently.

On the other hand, the static approach taken in SWeHG also has, of course, its limitations. First, static pages can not adapt their content dynamically to different user or patterns of usage. Second, dynamic systems can be connected more easily with other services providing additional functionality. Third, if the RDF repository, the rules, or the HTML templates change, the site has to be regenerated usually from scratch. Dynamic systems can adapt better to such changes. Fourth, if the RDF repository is large and many templates are used, then the number and size of generated pages can be large. In retrospect the limitations of the static, template based, publication approach outweighed the benefits, and the utilizing

and publishing of Semantic Web content on the Web went heavily on the dynamic side of the equation.

3.7 Contributions and Significance for Later Research

The rule-based facet projection and recommendation link generation solutions developed for *SWeHG* were later included in the *ONTODELLA* [108] facet projection and recommendation engine, which is part of the *OntoViews* [72] *FSS* framework. In this sense, the solutions developed initially for the *SWeHG* tool had a big impact on the development of *FSS*, and were later used in a large number of *FSS* applications such as *MuseumFinland* [52], *Orava* [65], *HealthFinland* [60], *Veturi* [81], and *SW-Suomi.fi* [95] to mention just a few.

The work presented in this chapter was also important in realizing the problems and shortcomings of the faceted search paradigm in general, and faceted *semantic* search in particular. These problems are detailed and explained in the next chapter, and the rest of the dissertation is dedicated to finding solutions to these problems.

4 Problems of Crisp Faceted Semantic Search

In this chapter a number of fundamental problems of faceted semantic search are described. The rest of the dissertation then provides solutions and answers to these problems.

4.1 Lack of Capability to Model Uncertainty

Ontologies and ontology-based—i.e., semantic—knowledge bases often contain unmodeled and unaddressed uncertainty, which sometimes hinders the ability of the *FSS* application to provide high-quality search results. This follows from the fact that ontologies are based on crisp logic, but our world is inherently uncertain. Also the information system—i.e., the *FSS* application—might be a source of uncertainty. According to Semantic Web ontologies there are not degrees or probabilities of truth, but statements are modeled either as *true* or *false*. We will consider the following two kinds of uncertainty:

Uncertainty related to concepts of an ontology The world is full of uncertainty, and situations that are difficult to model using crisp ontological concepts. For example, consider the geographical area *Lapland*. *Lapland* is divided between the countries *Finland*, *Norway*, *Russia*, and *Sweden*. The amount of overlap between *Lapland* and each of the countries varies from country to country. For example, about 1/3 of the total area of *Lapland* is situated in *Finland*, and *Lapland* constitutes about 1/4 of the geographical area of *Finland*. On the other hand, only about 1/10 of the *Lapland* is situated in *Russia*, and *Lapland* covers only a tiny fraction of the geographical area of *Russia*. These degrees of overlap are difficult to express using the

standard Semantic Web ontology languages, such as RDFS and OWL. As a result, an *FSS* application which has a geographical facet projected from a geographical taxonomy might return as the first result a search item about *Lapland* to the selection *Russia* even though this might seem unintuitive to the user, and the result might even not be related to *Russia* in any way.

Uncertainty related to annotations A search item is either annotated using an ontological concept or it is not. The ontological languages do not provide semantics for weighted annotations. This might also result in unoptimal search results. Consider for example, a photograph of a group of people eating sausage next to a fire near a lake. On the other side of the lake one can barely notice a silhouette of a church. A diligent information specialist will annotate this photo using concepts such as *Group, Human, Sausage, Fire, Eating, Lake, Church*. As a result, if an *FSS* application was generated from the semantic database which contains this photograph, someone might search for photographs about churches, and will be surprised to get a photograph of people eating sausages as the first result.

4.2 Information Overflow: Lack of Ranking

Faceted search does not incorporate the notion of relevance. In faceted search, search items are either annotated using the categories or mapped onto them using logic rules. In both cases, the search result for a category selection is the crisp set of search items annotated to it or its subconcepts.

Thus, all search results are equally relevant, and this is also the reason that the photograph about sausages might appear as the first result in the list when searching for churches in the example of the previous section. However, ranking of search results is a core feature of information retrieval systems [20], and its significance is

emphasized in environments such as national library collections, national museum collections, or the Internet where the amount of searchable information is vast [30]. When the amount of information is large, it is very important to be able to give the most relevant, or most probably relevant search result in the beginning of the result list. For this to be possible, the search system should be able to produce accurate ranking of the result set, and faceted search systems are not capable of doing this. Proponents of faceted search might say, that the user can then refine the search using another facet, or a subcategory of the selected category. This is not, however, always an optimal solution, because it might also be the case that the user has already specified the search in the accuracy of choice, and it would be artificial to make a further refinement.

4.3 Lack of User-Centric Facets

Taxonomies or ontologies often consist of complicated professional concepts needed for accurate indexing and annotation of content. Ontologies are typically organized according to a formal division of the topics or based on an upper-ontology. This is beneficial because it enables automatic reasoning over the ontologies. However, such categorizations are not necessarily useful as search facets because they can be difficult to understand and too detailed from the end-users viewpoint.

In this case, the user needs a view to the content that is different from the machine's or indexer's viewpoint. However, current faceted systems do not differentiate between indexer's, machine's, and end-user's views. For example, the *HealthFinland* portal publishes health content to ordinary citizens. Much of the material used has been indexed using complicated medical terms and classifications, such as Medical Subject Headings¹¹ (MeSH). Since the end-user is not an expert of the domain and is not familiar with the professional terms used in the ontology, the hierarchical

¹¹<http://www.nlm.nih.gov/mesh/>

organization of the ontology is not suitable for formulating end-user queries or presenting the result set, but only for indexing and machine processing. For this reason, in *HealthFinland* it was decided to create separate end-user facets and map them onto the annotation ontologies. The next sections provide solutions for this kind of mapping.

Part II

FUZZY APPROACH

5 Introduction to the Fuzzy Approach

This part of the dissertation will present solutions to the problems presented in Chapter 4 based on fuzzy set theory. Fuzzy set theory was chosen as a solution approach, because it is a widely used formalism to capture imprecise knowledge, and it has been applied successfully to many contexts in which uncertain or imprecise reasoning is required [64, 115, 27]. Specific to the semantic search context of this dissertation, fuzzy sets have been used to extend Semantic Web ontologies and description logic formalisms underlying these ontologies to provide uncertain reasoning [97, 96, 78], and to enhance query answering and semantic search [114, 16, 110]. A short review of this work will be given in Section 6.7 after *fuzzy FSS* is presented.

The problems addressed and solutions provided by the *fuzzy approach* are outlined in Table 5.1. This chapter will provide a short introduction to fuzzy sets, in order to equip the reader with required preliminary knowledge to understand the *fuzzy FSS* framework presented in Chapter 6. It will be seen that the fuzzy set theory is essentially an extension of the classical crisp set theory. Chapter 6 will present the *fuzzy faceted semantic search (FFSS)* framework. *FFSS* is based on weighted annotations, and supports the mapping of separate end-user facets onto annotation ontologies. Chapter 7 will present a method to produce weighted annotations by integrating the widely used *term frequency - inverse document frequency (TF-IDF)* weighting [93] with *FFSS*. Chapter 8 will summarize the contributions of the work presented in this part, and raise some remaining problems that will be then answered in the *Probabilistic Approach* part of the dissertation.

Table 5.1: The problems addressed and the solutions provided by *fuzzy FSS*.

	Problem	Solution	Chapter
1.	Semantic Web ontologies lack of support for modeling uncertainty inherent in the world, including the <i>FSS</i> system itself. This shortcoming hinders the system's ability to provide high quality search results for the user.	Annotations are weighted; i.e., fuzzy. Annotations can be weighted algorithmically by an ontological extension to the <i>TF-IDF</i> weighting method.	6., 7.
2.	The crisp faceted semantic search lacks the capability to rank search results based on relevance.	The degrees of relevance can be determined and search results ranked accordingly.	6.
3.	Concepts of annotation ontologies are not always suitable to be presented as search categories on the search GUI.	Distinct end-user's facets to search items can be created and mapped fuzzily onto indexing ontologies and the underlying search items (documents). Boolean combinations of annotation concepts can be used in the mappings.	6.

5.1 Crisp Sets

To indicate that an individual object x is a *member* of a set A , we write $x \in A$. The opposite case, i.e., that x is not a member of A , is written $x \notin A$. The universal set, denoted by U , is the set of all individual objects of concern in a particular universe from which sets can be formed. A set is defined by a *characteristic function*, that declares which elements of U are members of the set and which are not. Set A is defined by its characteristic function X_A as follows:

$$X_A(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases} \quad (5.1)$$

The Boolean set operations union, intersection and complement are defined by their characteristic functions as follows:

Union

$$X_{A \cup B}(x) = \max(X_A(x), X_B(x)), \forall x \in U \quad (5.2)$$

Intersection

$$X_{A \cap B}(x) = \min(X_A(x), X_B(x)), \forall x \in U \quad (5.3)$$

Complement

$$X_{\neg A}(x) = 1 - X_A(x), \forall x \in U \quad (5.4)$$

For each $x \in U$, when $X_A(x) = 1$, x is a member of A ; when $X_A(x) = 0$, x is a non-member of A . Thus, for each possible set A , each $x \in U$ is either a member of A or a nonmember of A , and there are no degrees of membership; i.e., the value of $X_A(x)$ can not be something in between 0 and 1.

5.2 Fuzzy Sets

The characteristic function of any crisp set A can be generalized such that the value assigned to each individual $x \in U$ fall within the range $[0, 1]$ and indicates the membership grade of these elements in the set A , such that higher values denote higher degrees of membership. Such a function is called a *membership function*, and the set defined by it is a fuzzy set. The membership function, thus, maps members of a universal set U , which is still a crisp set, to a real number in the range $[0, 1]$. This function is denoted

$$\mu_A : U \rightarrow [0, 1]. \quad (5.5)$$

The following, so called *standard fuzzy set operations*, generalize the three basic operations on crisp sets introduced above—the complement, union, and intersection—to fuzzy sets:

Standard complement

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \quad \forall x \in U \quad (5.6)$$

Standard union

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \quad \forall x \in U \quad (5.7)$$

Standard intersection

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \quad \forall x \in U \quad (5.8)$$

As can be seen, the standard fuzzy operations perform precisely as the corresponding operations for crisp sets, as defined by the *characteristic function*. These are, however, not the only possible generalizations of the crisp set operations. For each of the three operations there exists a broad class of functions whose members qualify as fuzzy generalizations of the classical operations. Thus, when adapting fuzzy sets to a specific application, it is possible to choose or determine the fuzzy operations that best fit to the context of the application. This greatly enhances the usefulness and applicability of fuzzy set theory. To qualify as a fuzzy generalization of a crisp set operation, the generalization must possess certain properties that intuitively guarantee that the fuzzy operation indeed is a generalization of the crisp set operation it claims to generalize. The properties that each fuzzy generalization operation must satisfy are defined as a set of axioms. As an example, we show next the basic definition of the properties that a function has to satisfy in order to qualify as a fuzzy complement operation.

Let $\mu_A(x)$ be the degree to which $x \in U$ belongs to the fuzzy set A . Let cA denote the fuzzy complement of A . Then $\mu_{cA}(x)$ can be interpreted both as the degree to which x belongs to cA , and as the degree to which x does not belong to A . The complement cA is defined by a function $c : [0, 1] \rightarrow [0, 1]$, which assigns a value $c(\mu_A(x))$ to each membership degree $\mu_A(x)$ of any given fuzzy set A . The value

$c(\mu_A(x))$ is interpreted as the value of $\mu_{cA}(x)$; i.e., $c(\mu_A(x)) = \mu_{cA}(x)$. To qualify as a fuzzy complement, any function c must satisfy at least the following two axioms:

Axiom c1

$$c(0) = 1 \text{ and } c(1) = 0 \quad (5.9)$$

Axiom c2

$$\forall a, b \in [0, 1], \text{ if } a \leq b, \text{ then } c(a) \geq c(b) \quad (5.10)$$

As can be seen, *Axiom c1* guarantees that c produces correct complements for crisp sets, and *Axiom c2* guarantees that when a membership grade in A increases by changing x , the corresponding membership grade in cA does not increase as well. Similar axioms are defined for the other fuzzy set operations as well, and in *fuzzy FSS* these will be used.

6 Fuzzy Faceted Semantic Search

This chapter is largely based on the article:

Markus Holi and Eero Hyvönen. 2006. Fuzzy View-Based Semantic Search [46]. In: Proceedings of the Asian Semantic Web Conference (ASWC2006).

This chapter presents a fuzzy version of the faceted semantic search paradigm, called *fuzzy faceted search (FFSS)* and will provide solutions to the problems of *crisp FSS* raised in Chapter 4. Table 5.1 summarizes the problems addressed and the solutions provided in this chapter.

The lack of support for modeling uncertainty in Semantic Web ontologies is given a partial solution by creating a formalism to weight annotations, and to reason based on these annotation weights in the context of *FSS*. The weighting of annotations is then used as a basis for computing the degrees of relevance of documents in relation to faceted search queries. *Fuzzy FSS* also includes a facility for the creation of distinct end-user's views—i.e., facets—to search items, and for the mapping of these facets onto the annotation ontologies and the underlying search items (documents).

The *fuzzy FSS* framework presented here generalizes *crisp FSS (CFSS)* from using crisp sets—as described in Section 2.2—to fuzzy set theory. In the following, this scheme is first developed using examples from the *HealthFinland* portal content. After this an implementation of the system is presented. Chapter 7 will then provide a method to create weighted annotations that can be utilized in *FFSS*. The weighting is done algorithmically based on textual content of the annotated documents. Chapter 8 will summarize the contributions and remaining problems of the fuzzy approach.

6.1 Architecture of the Framework

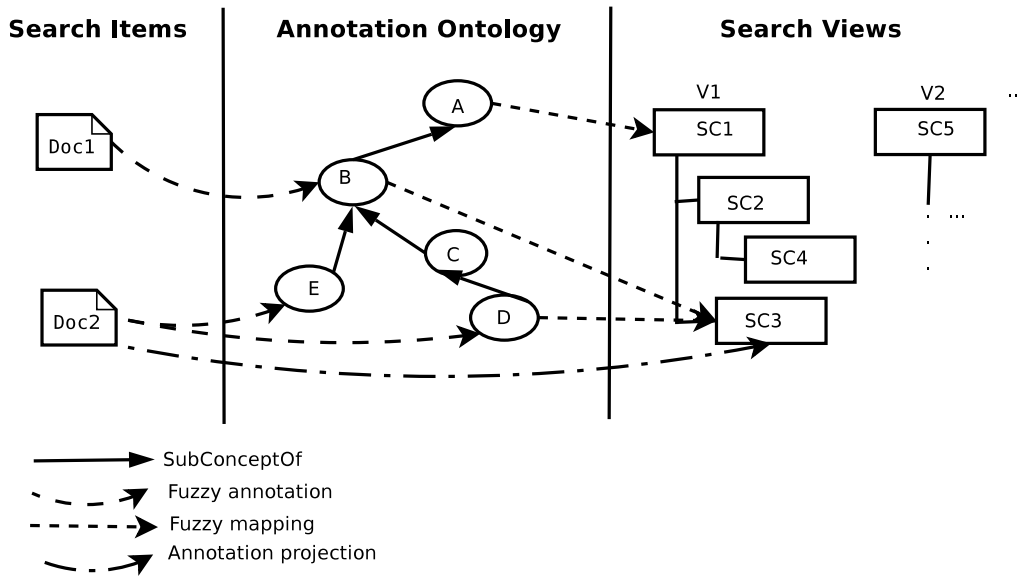


Figure 6.1: Components of the fuzzy faceted semantic search (FFSS) framework.

Figure 6.1 depicts the architecture of the fuzzy faceted semantic search framework. Next, we describe the components of the framework in short, and show how the set-theoretic interpretation of crisp faceted search is extended to fuzzy sets:

Search Items The search items are a finite set of documents D depicted on the left. As in *CFSS*, also in *FFSS* D is the set of individual objects from which sets are formed. Thus, D is the universal set of the fuzzy faceted search framework.

Annotation Ontology As in *CFSS* the search items are annotated according to the ontology. This might happen either automatically, semi-automatically, or manually by a human indexer. The ontology consists of two parts:

First, a finite set of annotation concepts AC . Recall from part I that according to the set-theoretic interpretation of *CFSS*, an annotation concept is seen as a set of documents. This is because in Section 2.2 search categories were

defined as sets of search items, and in *CFSS* annotation concepts are used as search categories. In *FFSS* this set theoretic interpretation of annotation concepts is extended to fuzzy sets; i.e., each annotation concept AC_j is seen as a fuzzy subsets of D . Intuitively, the degree of membership of a document in an annotation concept represents the relevance of the document to the annotation concept; i.e., how relevant the document would be to an individual interested in the annotation concept. Annotation concepts $AC_i \in AC$ are atomic; i.e., not defined as e.g. Boolean combinations of other concepts.

Second, the annotation ontology consists of subsumption relations between the annotation concepts, constituting a concept hierarchy. In different annotation ontologies these subsumption relations may be defined using different properties, such as *subClassOf*, *partOf*, or *broaderTerm*, depending on the used ontology language (RDFS, OWL, SKOS, etc.), and the subject domain in question. Whatever the exact properties used, in the context of *FFSS* these subsumption relations are interpreted as concept inclusion axioms $AC_i \subseteq AC_j$ ¹², where $AC_i, AC_j \in AC$ are annotation concepts, and $i, j \in N$, such that N is the set of natural numbers, and $i \neq j$. Recall, that in Section 2.2 subcategory relationships between search categories were interpreted as the subset relation. Thus, also in this point *FFSS* extends the crisp set theoretic interpretation of *CFSS* to fuzzy sets. Intuitively, the subset relation between the annotation concepts means, that if a document is relevant to a subconcept of AC_j it has to be relevant also to AC_j . Notice that we do not aim to redefine the semantics of the various subsumption properties used in ontologies, but only to give them a set theoretic meaning in the context of faceted search.

Facets As in *CFSS*, also in *FFSS* facets are sets of hierarchically organized search categories for the end-user to use during searching. In contrast to *CFSS*, in *FFSS* the facets are created and organized with end-user interaction in mind

¹²Subset relation between fuzzy sets is defined as: $AC_i \subseteq AC_j$ iff $\mu_{AC_j}(D_k) \geq \mu_{AC_i}(D_k)$, $\forall D_k \in D$, where D is the universal set.

and may be non-identical to the annotation concepts. As a natural extension to the set-theoretic interpretation of faceted search of Section 2.2, in *FFSS* each search category SC_i is a fuzzy subset of D .

Recall from 2.2, that in *CFSS* the intersection of documents related to selected search categories is returned as the result set. In *FFSS*, the intersection is replaced by the fuzzy intersection.

Search items related to a search category SC_i can be found by mapping them first onto annotation concepts by annotations, and then by mapping annotation concepts to SC_i . The search result S is not a crisp set of search items $S = SC_1 \cap \dots \cap SC_n = \{Doc_1, \dots, Doc_m\}$ as in *CFSS*, but a fuzzy set where the relevance of each item is specified by the value of the membership function mapping:

$$S = SC_1 \cap \dots \cap SC_n = \{(Doc_1, \mu_1), \dots, (Doc_m, \mu_m)\} \quad (6.1)$$

In the following the required mappings are described in detail.

6.2 Fuzzy Annotations

Search items (documents) have to be annotated in terms of the annotation concepts—either manually or automatically by using e.g. logic rules. Recall from Section 2.2, that in *CFSS* an annotation of a search item to a search category is interpreted as an explicit statement of the membership of the search item in the set of search items defining that category. In Figure 6.1, annotations are represented using bending dashed arcs from *Search Items* to *Annotation Ontology*. For example, in *CFSS* the dashed arc between *Doc1* and *B* would be interpreted as an explicit statement of the membership of *Doc1* in the set of search items defining *B*, in effect, $Doc1 \in B$.

In our approach, the relevance of a document with respect to an annotation concept may vary. This represents the fact that for a user interested in the concept B in Figure 6.1, $Doc1$ might be more or less relevant than $Doc2$ even though both of these documents are members of the set defining B . This representation of degrees of relevance is achieved by extending the crisp set interpretation of an annotation from the assertion of membership to the assertion of a degree of membership. For example, the dashed arc between $Doc1$, and B in Figure 6.1 is interpreted as $A_{\mu_B(Doc1)} \in [0, 1]$. Notice, that we do not interpret the fuzzy annotations directly as degrees of membership $\mu_B(Doc1)$, but only as assertions of degree of membership. This is because in order to know the degree of membership of the document D_i in the annotation concept AC_j we have also to take into account the degree of membership of D_i in the subconcepts of AC_j as described below.

Based on the fuzzy annotations, the membership function of each fuzzy set $AC_j \in AC$ can be defined. This is done based on the meaning of subsumption, i.e. inclusion. One concept is subsumed by the other if and only if all individuals in the set denoting the subconcept are also in the set denoting the superconcept; i.e., if being in the subconcept implies being in the superconcept [90]. In terms of fuzzy sets this means that $AC_i \subseteq AC_j$, and $\mu_{AC_i}(D_i) = \nu$ implies that $\mu_{AC_j}(D_i) \geq \nu$, where $\nu \in (0, 1]$, D_i is a search item, and $\mu_{AC_i}(D_i)$ and $\mu_{AC_j}(D_i)$ are the membership functions of sets AC_i and AC_j , respectively.

Thus, we define the membership degree of a document D_i in AC_j as the maximum of its concept membership assertions made for the subconcepts of AC_j (including AC_j) itself.

$$\forall D_i \in D, \mu_{AC_j}(D_i) = \max(A_{\mu_{AC_i}(D_i)}), \text{ where } AC_i \subseteq AC_j \quad (6.2)$$

For example, assume that we have a document $D1$ that is annotated by the annota-

tion concept *Asthma* with weight 0.8, i.e. $A_{\mu_{Asthma}(D1)} = 0.8$, and that *D1* is also annotated with weight 0.1 to the annotation concept *Diseases*; i.e., $A_{\mu_{Diseases}(D1)} = 0.1$. Assume further, that in the annotation ontology *Asthma* is a subconcept of *Diseases*, i.e. $Asthma \subseteq Diseases$. Then,

$$\mu_{Diseases}(D1) = \max(A_{\mu_{Asthma}(D1)}, A_{\mu_{Diseases}(D1)}) = \max(0.8, 0.1) = 0.8 \quad (6.3)$$

6.3 Fuzzy Mappings

Each search category SC_i in a facet V_j is defined using concepts from the annotation ontology by a finite set of fuzzy concept inclusion axioms that we call *fuzzy mappings*. M is the set of all fuzzy mappings in the FFSS system, and each fuzzy mapping M_k is defined as:

$$M_k = AC_i \subseteq_{\nu} SC_j, \text{ where } AC_i \in AC, SC_j \in V_l, i, j, k, l \in N \quad (6.4)$$

where N is the set of natural numbers, and $\nu \in (0, 1]$

M_k constrains the meaning of a search category SC_j by telling to what degree ν the membership of a document D_m in an annotation concept AC_i implies its membership in SC_j .

Thus, fuzzy inclusion is interpreted as fuzzy implication. The definition is based on the connection between inclusion and implication described previously. This is extended to fuzzy inclusion as in [97, 25]. We use Goguen's fuzzy implication, i.e.

$$i(\mu_{AC_j}(D_k), \mu_{SC_l}(D_k)) = 1 \text{ if } \mu_{SC_l}(D_k) \geq \mu_{AC_j}(D_k), \text{ and} \quad (6.5)$$

$$\mu_{SC_l}(D_k)/\mu_{AC_j}(D_k) \text{ otherwise, } \forall D_k \in D$$

Applying this fuzzy implication, each fuzzy mapping $M_k = AC_i \subseteq_{\nu} SC_j$ defines a fuzzy set $SC_j^{M_k}$ s.t. $\mu_{SC_j^{M_k}}(D_l) = \nu * \mu_{AC_i}(D_l), \forall D_l \in D$, where $i(\mu_{AC_i}(D_l), \mu_{SC_j}(D_l)) = \nu$ and $\nu \in (0, 1]$. Apart from its semantic compatibility, Goguen's implication was chosen, because it provides a straight-forward formula to compute the set $SC_j^{M_k}$.

In *FFSS* a search category SC_j is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. This is called the *union principle* in *FFSS*, it is illustrated in Figure 6.2(f), and a concrete example of the application of this principle will be given in Section 6.4 after fuzzy mappings to Boolean combinations of annotation concepts will be described. Using Gödel's union function¹³ the membership function of SC_j is

$$\mu_{SC_j}(D_i) = \max(\mu_{SC_1}(D_i), \dots, \mu_{SC_n}(D_i), \mu_{SC_j^{M_1}}(D_i), \dots, \mu_{SC_j^{M_m}}(D_i)), \forall D_i \in D \quad (6.6)$$

where $SC_{1,\dots,n}$ are subcategories of SC_j , and $M_{1,\dots,m}$ are the fuzzy mappings pointing to SC_j . This extends the idea of faceted search, where search categories correspond directly to annotation concepts.

Let us continue with the example case in the end of Section 6.2 where we defined the membership of document D_1 in the annotation concept *Diseases*. If we have a fuzzy mapping

$$M_1 = \textit{Diseases} \subseteq_{0.1} \textit{Food\&Diseases} \quad (6.7)$$

¹³ $\mu_{A \cup B}(D_i) = \max(\mu_A(D_i), \mu_B(D_i)), \forall D_i \in D$

then—if this is the only mapping that the search category *Food & Diseases* has, and *Food & Diseases* does not have subcategories—the membership degree of the document D_1 in *Food&Diseases* is

$$\mu_{Food\&Diseases}(D_1) = \mu_{Diseases}(D_1) * 0.1 = 0.8 * 0.1 = 0.08 \quad (6.8)$$

Intuitively, the fuzzy mapping reveals to which degree the annotation concept can be considered a subconcept of the search category. Fuzzy mappings can be created by a human expert or by an automatic or a semi-automatic ontology mapping tool. In Figure 6.1, fuzzy mappings are represented using straight dashed arcs.

The fuzzy mappings of a search category can be *nested*. Intuitively, in this case a search category is explicitly mapped to both an annotation concept and one of its subconcepts, as depicted in Figure 6.2(e). Formally, two fuzzy mappings M_1 and M_2 are *nested* if

$$\begin{aligned} M_1 &= AC_i \subseteq_{\kappa} SC_i, \\ M_2 &= AC_j \subseteq_{\nu} SC_i, \text{ and} \\ AC_i &\subseteq AC_j. \end{aligned} \quad (6.9)$$

To avoid the double mapping of the subconcept—through the explicit mapping, and through the mapping of the superconcept—we dissolve this nesting. In effect, we modify the fuzzy mappings so that documents will be projected to the search category only through the most specific mapping. To achieve this, nesting between the fuzzy mappings M_1 and M_2 above is interpreted as a shorthand for

$$M_1 = AC_i \subseteq_{\mu} SC_i \text{ and } M_2 = (AC_j \cap \neg AC_i) \subseteq_{\nu} SC_i. \quad (6.10)$$

This interpretation actually dissolves the nesting. For example, if we have mappings

$$\begin{aligned} M_1 &= \textit{Animal nutrition} \subseteq_{0.1} \textit{Nutrition}_{sc}, \textit{ and} \\ M_2 &= \textit{Nutrition} \subseteq_{0.9} \textit{Nutrition}_{sc} \end{aligned} \quad (6.11)$$

and in the annotation ontology

$$\textit{Animal nutrition} \subset \textit{Nutrition}, \quad (6.12)$$

then M_1 is actually interpreted as

$$M_1 = \textit{Nutrition} \cap \neg \textit{Animal nutrition} \subseteq_{0.9} \textit{Nutrition}_{sc}. \quad (6.13)$$

In some situations it is useful to be able to map a search category to a Boolean combination of annotation concepts. For example, if a facet contains the search category *Food & Exercise* then those documents that talk about both nutrition and exercise are relevant. Thus, it would be valuable to map *Food & Exercise* to the intersection of the annotation concepts *Nutrition* and *Exercise*. To enable mappings of this kind, a Boolean combination of annotation concepts can be used in a fuzzy mapping. The Boolean combinations are:

1. $AC_1 \cap \dots \cap AC_n$ (intersection),
2. $AC_1 \cup \dots \cup AC_n$ (union), and
3. $\neg AC_1$ (negation),

where $AC_1, \dots, AC_n \in AC$.

In the next section, a detailed description is presented on how to determine the fuzzy sets corresponding to search categories in each of the Boolean cases. The real-world cases of Figure 6.2 will be used as examples in the text. In Section 6.5 we describe how to execute the faceted search based on the projected annotations and the end-user's selections.

6.4 Mappings to Boolean Concepts

In the following, the membership function definition for each type of Boolean concept is listed, according to the widely used Gödel's functions¹⁴:

Union Case $AC_j = AC_l \cup \dots \cup AC_n$: The membership degree of a document in AC_j is the maximum of its concept membership values in any of the components of the union concept:

$$\forall D_k \in D, \mu_{AC_j}(D_k) = \max(\mu_{AC_i}(D_k)), \text{ where } i \in l, \dots, n. \quad (6.14)$$

In the example union case of Figure 6.2(c) we get

$$\begin{aligned} & \mu_{\text{Thinness} \cup \text{Obesity}}(D5) \\ &= \max(\mu_{\text{Thinness}}(D5), \mu_{\text{Obesity}}(D5)) \\ &= \max(0, 0.8) = 0.8. \end{aligned} \quad (6.15)$$

Intersection Case $AC_j = AC_l \cap \dots \cap AC_n$: The membership degree of a document in AC_j is the minimum of its concept membership values in any of the components of the union concept:

$$\forall D_k \in D, \mu_{AC_j}(D_k) = \min(\mu_{AC_i}(D_k)), \text{ where } i \in l, \dots, n. \quad (6.16)$$

In the example intersection case of Figure 6.2(b) we get

$$\begin{aligned} & \mu_{\text{Nutrition} \cap \text{Exercise}}(D1) \\ &= \min(\mu_{\text{Nutrition}}(D1), \mu_{\text{Exercise}}(D1)) \\ &= \min(0.4, 0.3) = 0.3. \end{aligned} \quad (6.17)$$

¹⁴If A and B are fuzzy sets of the universal set X , then $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$, $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$, and $\mu_{\neg A}(x) = 0$, if $\mu_A(x) > 0$, 1 otherwise, $\forall x \in X$. Notice, that the Gödel complement differs from the standard complement introduced in Section 5.2. It, however, complies with the axioms of fuzzy complement introduced in Section 5.2, and represents well the intuition of semantic faceted search that documents not annotated at all to a concept belong to its complement, and other documents—no matter how small the membership degree—are still considered members.

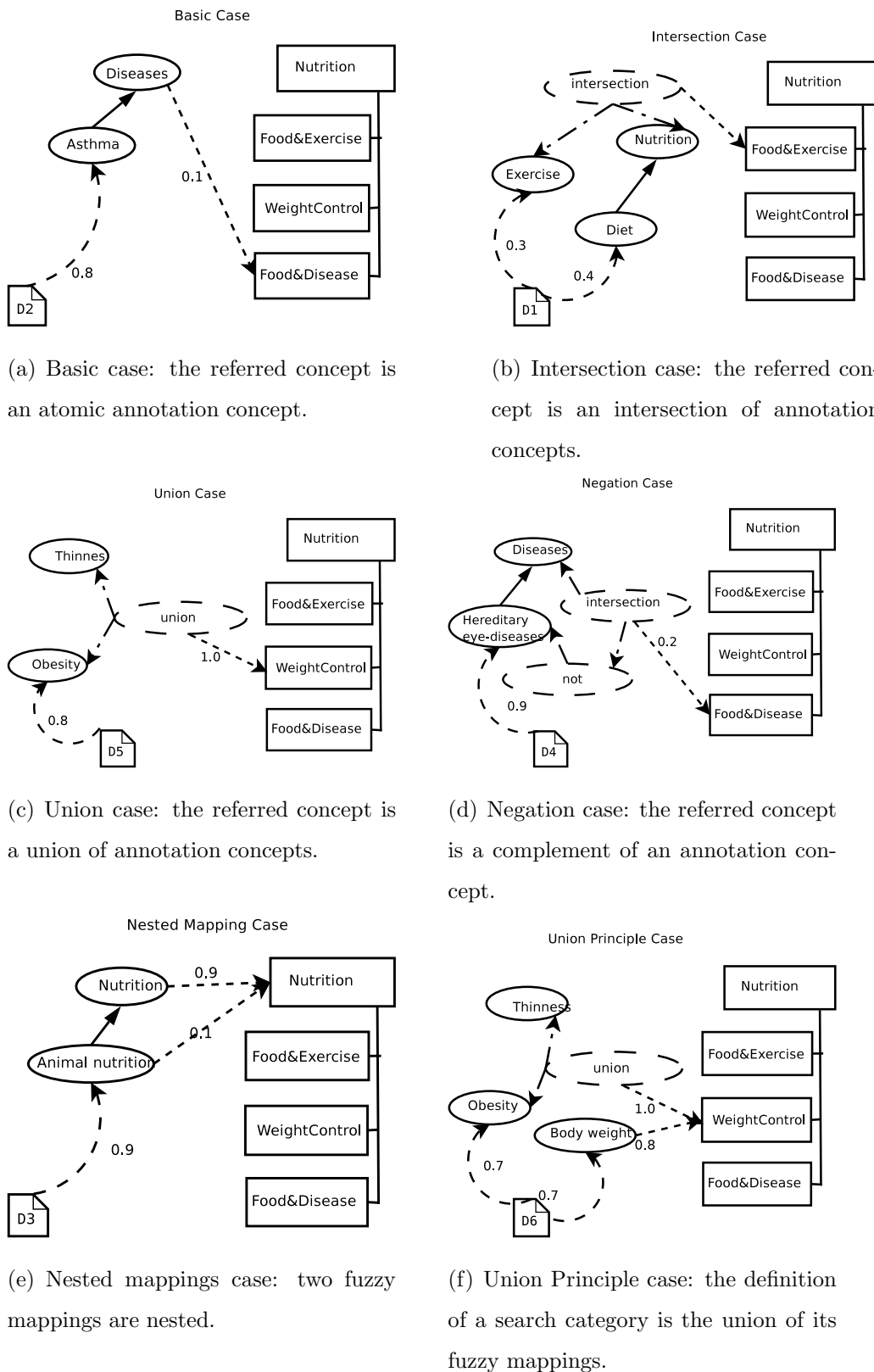


Figure 6.2: Real-world examples of annotation projection cases

Negation Case $AC_j = \neg AC_l$: The membership degree of a document in AC_j is 1 if the membership degree of the document in AC_k is 0, and 0 otherwise.

$$\begin{aligned} \forall D_k \in D, \mu_{AC_j}(D_k) = 0 \text{ if } \mu_{AC_l}(D_k) > 0 \text{ and} \\ \mu_{AC_j}(D_k) = 1 \text{ if } \mu_{AC_l}(D_k) = 0. \end{aligned} \quad (6.18)$$

In the example negation case of Figure 6.2(d) we get

$$\mu_{\neg Hereditary \ eye-diseases}(D4) = 0, \quad (6.19)$$

because $(\mu_{Hereditary \ eye-diseases}(D4) = 0.9) > 0$.

Notice, that in Figure 6.2(d), the search category *Food and Diseases* is mapped to the intersection of *Diseases* and $\neg Hereditary \ eye-diseases$, instead of directly mapping to the negation concept. This is because it seems that in real-world situations search categories are often not mapped directly to negations of concepts, but negation is often used as a component of another Boolean concept. In fact, we could not find any good example of mapping of a search category directly to a negation concept.

After the membership function of each Boolean concept is defined, the membership functions of the search categories can be computed based on the fuzzy mappings. This is done using the *union principle* described in Section 6.3. For example, in Figure 6.2(f) the projection of document *D6* to the search category *Weight Control* is done by first computing the membership degree of *D6* in the relevant annotation concepts:

$$\begin{aligned} \mu_{Thinness \cup Obesity}(D6) &= \max(\mu_{Thinness}(D6), \mu_{Obesity}(D6)) = \max(0, 0.7) = 0.7 \\ \mu_{Body \ weight}(D6) &= 0.7 \end{aligned} \quad (6.20)$$

Now, the fuzzy mapping M_1 with weight $\nu_1 = 1$ between the union concept *Thinness* \cup *Obesity* and *Weight Control* yields the membership degree

$$\mu_{M_1}(D6) = \mu_{Thinness \cup Obesity}(D6) * \nu_1 = 0.7 * 1 = 0.7. \quad (6.21)$$

The fuzzy mapping M_2 with weight $\nu_2 = 0.8$ between the annotation concept *Body weight* and *Weight Control* yields the membership degree

$$\mu_{M_2}(D6) = \mu_{Weight\ control}(D6) * \nu_2 = 0.7 * 0.8 = 0.56. \quad (6.22)$$

Because each search category is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it, and *Weight Control* does not have any subcategories, we get

$$\mu_{WeightControl}(D6) = \max(\mu_{M_1}(D6), \mu_{M_2}(D6)) = \max(0.7, 0.56) = 0.7. \quad (6.23)$$

6.5 Performing the Search

In faceted search the user can query by choosing at most one category from each facet. Recall from Section 2.2 that in CFSS a search is defined as the intersection of the result sets corresponding to the selected search categories; i.e., $S = \bigcap SC_i$, $\forall i \in 1, \dots, k$, where the user has selected the search categories SC_1, \dots, SC_k .

In *FFSS* we extend the crisp intersection to fuzzy sets. Using Gödel's intersection [115], we have:

$$\mu_S(D_j) = \min(\mu_{SC_1}(D_j), \dots, \mu_{SC_k}(D_j)), \quad \forall D_j \in D. \quad (6.24)$$

As a result, search results can be sorted according to relevance in a well-defined manner, based on the values of the membership function.

6.6 Implementation

In the following an implementation of our framework is presented. In Sections 6.6.1 and 6.6.2, RDF [7] representations of fuzzy annotations and facets are described, respectively. Section 6.6.3 presents an algorithm for the annotation projection discussed in Section 6.4.

6.6.1 Representing Fuzzy Annotations

We comply with Semantic Web standards, we created an RDF representation for fuzzy annotations. In the representation each document is a resource represented by an URI, which is the URL of the document. The fuzzy annotation of the document is represented as an instance of a 'Descriptor' class with two properties. 1) A 'describes' property points to a document URI, and 2) a 'hasElement' property points to a list representing the fuzzy annotations. The fuzzy annotation is an instance of a 'DescriptorElement' class. This class has two properties: 1) 'hasConcept' which points to the annotation concept, and 2) 'hasWeight', which tells the weight; i.e., the fuzziness of the annotation. For example, the fuzzy annotation of the document *D1* in Figure 6.2 is represented in the following way.

```
<!-- instances of DescriptorElement represents fuzzy annotations-->
<DescriptorElement rdf:ID="descriptorelement_63">
  <!-- hasTerm points to the concept to which a document is annotated,
  here it is a MeSH term.-->
  <hasTerm rdf:resource="&mesh;D004032"/>
  <!-- hasWeight tells the degree of membership of the document in the fuzzy set
  that represents the concept which hasTerm points to-->
  <hasWeight>0.4</hasWeight>
</DescriptorElement>
<DescriptorElement rdf:ID="descriptorelement_64">
  <hasTerm rdf:resource="&mesh;D015444"/>
  <hasWeight>0.3</hasWeight>
</DescriptorElement>
<!-- Instance of descriptor class defines the annotation of a document -->
<Descriptor rdf:ID="Descriptor_6">
```

```

<!-- describes points to the annotated document>
<describes rdf:resource="#D1"/>
<!-- The hasElement is a collection that contains the individual fuzzy
      annotations (DescriptorElement instances) of the document-->
<hasElement rdf:parseType="Collection">
  <DescriptorElement rdf:about="#descriptorelement_63"/>
  <DescriptorElement rdf:about="#descriptorelement_64"/>
</hasElement>
</Descriptor>

```

Also the projected annotations are represented in the same manner.

6.6.2 Representing Search Facets

We created an RDF representation of the facets and the mappings between the search categories of the facets and the annotation concepts. Our representation is based on the Simple Knowledge Organization System (SKOS) [10, 12]. We chose SKOS mainly because it provides readily defined properties for representing mappings between concepts. For example the search categories *Nutrition* and *Nutrition&Diseases* in Figure 6.2 are represented in the following way:

```

<skos:Concept rdf:ID="Nutrition">
  <skos:prefLabel xml:lang="en">Nutrition
</skos:prefLabel>
  <fuzzy:mapping>
    <rdf:Description>
      <skosMap:narrowMatch rdf:resource="&mesh;D009747"/>
      <fuzzy:degree>0.9</fuzzy:degree>
    </rdf:Description>
  </fuzzy:mapping>
  <fuzzy:mapping>
    <rdf:Description>
      <skosMap:narrowMatch rdf:resource="&mesh;D000824"/>
      <fuzzy:degree>0.1</fuzzy:degree>
    </rdf:Description>
  </fuzzy:mapping>
</skos:Concept>
<skos:Concept rdf:ID="FoodAndDisease">
  <skos:prefLabel xml:lang="en">Food and Disease
</skos:prefLabel>

```

```

<skos:broader rdf:resource="#Nutrition"/>
<fuzzy:mapping>
  <rdf:Description>
    <skosMap:narrowMatch>
      <skosMap:AND>
        <rdf:li rdf:resource="&mesh;Diseases"/>
        <rdf:li>
          <skosMap:NOT>
            <rdf:li rdf:resource="&mesh;D015785"/>
          </skosMap:NOT>
        </rdf:li>
      </skosMap:AND>
    </skosMap:narrowMatch>
    <fuzzy:degree>0.25</fuzzy:degree>
  </rdf:Description>
</fuzzy:mapping>
</skos:Concept>

```

We use the *narrowMatch* property of SKOS for the mapping because its semantics corresponds closely to the implication operator as we want: If a document d is annotated with an annotation concept AC_1 , and AC_1 is a *narrowMatch* of a search category SC_1 , then the annotation can be projected from AC_1 to SC_1 . The *degree* property corresponds to the degree of truth of the mapping used in SKOS.

6.6.3 Projection of Annotations

We implemented the projection of annotations—i.e., the computation of the membership degrees of the documents in each search category—using the Jena Semantic Web Framework¹⁵. The implementation performs the following steps:

1. The RDF data described above is read and a model based on it is created. This involves also the construction of the concept hierarchies based on the RDF files.
2. The nested mappings are handled as described in Section 6.3.

¹⁵<http://jena.sourceforge.net/>

3. The membership function of each annotation concept is computed using the method described in Section 6.2.
4. The membership function of each search category is computed using the method described in Section 6.3.

6.7 Related Work

We have applied the idea presented by Straccia [97] in his fuzzy extension to the description logic *SHOIN(D)* and Bordogna [25] of using fuzzy implication to model fuzzy inclusion between fuzzy sets. Recall, that in *FFSS* the concepts and search categories are modeled as fuzzy sets, and the fuzzy inclusion is used to model the mapping of search categories to annotation concepts. However, in contrast to the fuzzy extensions of description logics, *FFSS* operates under the closed world assumption. Thus, something is true if it is found in the current knowledge base, otherwise it is false. Under the open world assumption used in description logics something is false only if it is logically impossible; i.e., if it is not possible to construct a knowledge base in which the statement would be found true. Besides the fuzzy extension of Straccia, also other fuzzy extensions to description logic exist, such as [96, 78].

Zhang et al. [114] have applied fuzzy description logic and information retrieval mechanisms to enhance query answering in semantic portals. Their framework is similar to ours in that both the textual content of the documents and the semantic metadata is used to improve information retrieval. Chapter 7 will describe a method by which the textual content of documents can be used to weight annotations. However, the main difference in the approaches is that their work does not help the user in query construction whereas the work presented in this paper does this by providing an end-user specific view to the search items.

Akrivas et al. [16] present an interesting method for context sensitive semantic query expansion. In this method, the user’s query words are expanded using fuzzy concept hierarchies. An inclusion relation defines the hierarchy. The inclusion relation is defined as the composition of subclass and part-of relations. Each word in a query is expanded by all the concepts that are included in it according to the fuzzy hierarchy. In [16], the inclusion relation is of the form $P(a, b) \in [0, 1]$ with the following meaning: A concept a is completely a part of b . High values of the $P(a, b)$ function mean that the meaning of a approaches the meaning of b . In our work the relation itself—i.e., inclusion—is fuzzy, and not only the sets. This fuzzy inclusion was interpreted as fuzzy implication, meaning that the inclusion relation itself is partial. This interpretation enables the modeling of the vague or inexact relations; i.e., fuzzy mappings using fuzzy logics.

Widyantoro and Yen [110] have created a domain-specific search engine called PASS. The system includes an interactive query refinement mechanism to help to find the most appropriate query terms. The system uses a fuzzy ontology of term associations as one of the sources of its knowledge to suggest alternative query terms. The ontology is organized according to narrower-term relations. The ontology is automatically built using information obtained from the system’s document collections.

6.8 Summary

This chapter presented the *fuzzy faceted semantic search (FFSS)* framework, which extends the *CFSS* from crisp sets to fuzzy sets. In *FFSS* the annotation concepts and search categories are modeled as fuzzy sets of document instances. This fuzzyfication is based on weighted document annotations, such that the annotation weight indicates the degree of membership of the document in the fuzzy set represented by the concept. The membership degree represents the relevance of the document to the annotation concept.

FFSS supports the definition of separate end-user facets that are then mapped to the annotation concepts using fuzzy mappings. The end-user facets can be mapped to either individual annotation concepts or Boolean combinations of these annotation concepts. In the mappings fuzzy generalizations of crisp set operations are used. The used fuzzy set operations are *complement*, *intersection*, *union*, and *inclusion* (i.e., *subset*). *FFSS* is a generalization of *crisp FSS* in the sense that the crisp version is a special case of *FFSS* such that both the annotation weights, and the fuzzy mapping values are either 0 or 1. As the *FFSS* is dependent on the existence of weighted annotations it is important to be able to create such annotation weights easily. The next Chapter (7) presents a method to integrate *term frequency - inverse document frequency (TF-IDF)* weighting with annotations of documents to algorithmically produce such annotation weights.

Chapter 7 also provides a preliminary empirical evaluation of *FFSS* that was conducted in the year 2006 when the article on which the chapter is based was first published. A more complete evaluation of *FFSS*, is presented in Chapter 13. According to both of these evaluations the document rankings provided by *FFSS* significantly improve retrieval performance over *CFSS*.

7 Integrating Term Frequency - Inverse Document Frequency (TF-IDF) Weighting with Fuzzy Faceted Semantic Search

This chapter is largely based on the article:

Markus Holi, Eero Hyvönen, and Petri Lindgren. 2006. Integrating tf-idf Weighting with Fuzzy View-Based Search[48]. In: Proceedings of the ECAI Workshop on Text-Based Information Retrieval (TIR-06).

This chapter presents a method to weight existing crisp annotations using an ontological extension of the widely used *TF-IDF* term weighting scheme [93]. The fuzzy annotation weight reflects the relevance of the document to the annotation concept, and in the context of *FFSS* it can be used as the membership degree of the individual representing the document in the fuzzy set representing the annotation concept.

The method presented in this chapter is important because it presents an algorithmic way to weight existing crisp annotations. The manual weighting of annotations might prove to be in many cases too arduous a task. Furthermore, this same weighting method can be used also to produce the probabilistic annotations required by the *probabilistic faceted semantic search* framework that will be presented in Chapter 11.

In the following, we will first briefly describe *TF-IDF* weighting method, and then present the ontological extension to it. After the ontological extension of *TF-IDF* is developed, a test implementation and evaluation is presented.

7.1 The TF-IDF Method

The *TF-IDF* [93] weighting method is often used in information retrieval. It is a statistical technique to evaluate how important a term is to a document. The importance increases proportionally to the number of times a word appears in the document but is offset by how common the word is in all of the documents in the document collection. *TF-IDF* is often used by search engines to find the most relevant documents to a user's query. There are many different formulas used to calculate *TF-IDF*. A widely used formula that calculates a normalized *TF-IDF* weight¹⁶ is presented below. The formula gives values between 0 and 1.

The term frequency $tf_{t_i Doc_j}$ of term t_i in a document Doc_j gives a measure of the importance of the term within the document. In the formula that we used, $tf_{t_i Doc_j}$ is simply the number of occurrences of t_i in Doc_j .

The inverse document frequency *idf* is a measure of the general importance of the term. In the formula that we used, idf_{t_i} is the natural logarithm of the number of all documents N divided by df_{t_i} —the number of documents containing the term t_i , i.e.

$$idf_{t_i} = \log\left(\frac{N}{df_{t_i}}\right) \quad (7.1)$$

The normalized *TF-IDF* weight of the term t_i in document Doc_j is

$$tf-idf_{t_i Doc_j} = \frac{tf_{t_i Doc_j} * idf_{t_i}}{\sqrt{\sum_{i=1}^M (tf_{t_i Doc_j} * idf_{t_i})^2}} \quad (7.2)$$

¹⁶See, <http://www.sims.berkeley.edu:8000/courses/is202/f05/LectureNotes/202-20051110.pdf>

where M is the number of terms in Doc_j . A high weight in *TF-IDF* is reached by a high term frequency in the given document and a low document frequency in the whole collection of documents.

7.2 Ontological Extension of TF-IDF

We extended the *TF-IDF* weighting method so that it can be used to weight existing crisp document annotations. A crisp annotation of a document Doc_j is the set $A_{Doc_j} = \{AC_1, \dots, AC_n\}$, where AC_1, \dots, AC_n are concepts of the annotation ontology. The weighting is done based on the textual content of the document and the description of each concept AC_1, \dots, AC_n in the ontology.

The main idea is that instead of calculating the importance of each word to a given document Doc_j we calculate the importance of each concept in A_{Doc_j} to the document. The weighting of the annotation of Doc_j is done as follows:

1. A set of words W_{AC_i} is created for each concept in A_{Doc_j} . The set is the union of the labels of AC_i and the labels of the subconcepts of AC_i in the ontology.
2. The term frequency $tf_{AC_i Doc_j}$ for each AC_i in Doc_j is counted. This is done by reading (automatically) through Doc_j and each time that a word that belongs to the set W_{AC_i} is encountered $tf_{AC_i Doc_j}$ is increased by one. The counter starts from 1, thus if there are no occurrences of AC_i in Doc_j , then $tf_{AC_i Doc_j} = 1$. This is to recognize the fact that if a document is annotated using AC_i then AC_i is relevant to the document even if the content does not speak of AC_i directly.
3. The number of documents annotated with AC_i , i.e. df_{AC_i} is counted.

Now

$$idf_{AC_i} = \log\left(\frac{N}{df_{AC_i}}\right) \quad (7.3)$$

where N is the number of documents in the collection, and

$$tf-idf_{AC_i Doc_j} = \frac{tf_{AC_i Doc_j} * idf_{AC_i}}{\sqrt{\sum_{i=1}^M (tf_{AC_i Doc_j} * idf_{AC_i})^2}} \quad (7.4)$$

where M is the number of concepts in A_{Doc_j} .

The ontological extension of *TF-IDF* presented above offers some benefits when compared to traditional *TF-IDF*. The benefits are a result of the utilization of the structure of the annotation ontology. First, terms that are expressions of the same concept are detected. Thus they can be represented using a single concept identifier and the representation of the document content is compressed. Second, the concept hierarchies enable a better query answering. For example, the system knows that documents about dogs are relevant to a query about animals.

One possible pitfall here is that the recall will improve on the expense of precision as a result of this annotation expansion. The effects of ontological query expansion on recall-precision performance has been studied e.g. in [68, 67]. For an example of a situation where this could occur consider a database of maps which are annotated using a geographical ontology. A map could be annotated using e.g. Helsinki, the capital of Finland which in the geographical ontology would be a subconcept of Finland based on geographical inclusion. Now, as a result of the ontological *TF-IDF* the map about Helsinki would be counted relevant to Finland. For a user searching maps about Finland this might feel misleading, if on the search interface Finland is

explained to mean maps of Finland as a whole. However, this problem is overcome when the ontological *TF-IDF* is used as part of the *FFSS* framework, which enables the mapping of a search category to an annotation concept without including its subconcepts. The search system might define the search category *Finland* using a mapping to a Boolean concept that is the intersection of the annotation concept *Finland* and the negation of the union of its child concepts. In this case the map about Helsinki will not be included in the result set of the search category *Finland*.

7.3 Evaluation

To evaluate the method, we implemented the ontologies, annotations and search views using RDF. The algorithms were implemented using Java¹⁷ and its Semantic Web Framework Jena¹⁸.

Our document set consisted of 163 documents of the website of the National Public Health Institute¹⁹ of Finland (NPHI).

As an annotation ontology we created a SKOS translation of FinMeSH [5], the Finnish translation of MeSH [4]. The fuzzy annotations were created in two steps. First, an information scientist working for the NPHI annotated each document with a number of FinMeSH concepts. These annotations were crisp. Second, the crisp annotations were weighted using an ontological version of the *TF-IDF* weighting method. We scanned through each document and weighted the annotations based on the occurrences of the annotation concept labels (including subconcept labels) in the documents. The weight was then normalized, to conform to the fuzzy set representation. The normalized version of *TF-IDF* naturally translates into a membership degree in a fuzzy set as defined in *FFSS*. In *TF-IDF* the value is between

¹⁷<http://java.sun.com>

¹⁸<http://jena.sourceforge.net>

¹⁹See, <http://www.ktl.fi/>

0 and 1 as a membership degree in a fuzzy set. 0 means that the document does not match the concept at all which can be interpreted as a membership degree 0 in the fuzzy set defined by the concept, and 1 means that the document matches the concept perfectly which can be interpreted as a membership degree 1 in the respective fuzzy set.

The search facets with mappings to annotation concepts were designed and created by hand, and five search categories were chosen randomly. These categories were: *Diabetes*, *Food*, *Food Related Diseases*, *Food Related Allergies*, and *Weight Control*. The document set of each category was divided into two parts. The first part consisted of the documents whose rank was equal or better than the median rank, and the second part consisted of documents below the median rank. Then a document was chosen from each part randomly. Thus, each of the chosen categories was attached with two documents, one representing a well ranking document, and the other representing a poorly ranking document.

The test group consisted of five subjects, each of which was asked to read the two documents attached to a search category, e.g. *Diabetes*, in a random order, and pick the one that they thought was more relevant to the search category. This was repeated for all the selected search categories. Thus, each tested person read 10 documents. The relevance assessment of the test subjects were compared to the ordering done by our implementation. According to the results every test subject ordered the documents in the same way that the algorithm did. Due to the small number of test subjects this evaluation has to be taken as preliminary. Also the document selection method—where only two documents were selected for each search category—only gives a rough estimation of whether the method produces good rankings. However, the results strongly suggest that *FFSS* used together with the ontological extension to *TF-IDF* produces useful ranking of document results when compared to the crisp faceted semantic search, which does not support ranking based on content relevance at all.

8 Contributions and Lessons Learned

This chapter will summarize the contributions, and lessons learned of the work presented in this part (*Fuzzy Approach*) of the dissertation. Some problems are then raised, which will be solved as part of the work presented in the next part (*Probabilistic Approach*).

8.1 Contributions

The presented *FFSS* method provides the following solutions to the problems presented in Chapter 4.

Modeling of Annotation Related Uncertainty *FFSS* supports, and in fact, is based on weighted—i.e., fuzzy—annotations. It provides a formalisms to represent and reason based on weighted annotations, where the annotation weight is interpreted as a degree of membership of the annotated search item in the fuzzy set representing the annotation concept. This degree of membership is interpreted as the relevance of the search item to the annotation concept. Thus, *FFSS* presents a solution to the annotation specific uncertainty problem presented in Chapter 4.

Furthermore, Chapter 7 presented a method by which crisp annotations can be algorithmically weighted, increasing the applicability of *FFSS*. This, method is an ontological extension of the *TF-IDF* weighting method, and it provides also some benefits when compared to traditional *TF-IDF* weighting. In this comparison, the main benefits of ontological *TF-IDF* are the following: First, terms that are expressions of the same concept can be represented using a single concept identifier which results in a compressed representation

of the document content. Second, the concept hierarchies of the ontologies can be utilized to enable better query answering as a result of the search system becoming aware of the semantic relations between concepts.

Ranking of the result set Crisp faceted semantic search provides sophisticated means to order results by grouping. However, it does not provide ways to rank results. By extending the set theoretical model of view-based search to fuzzy sets, ranking the results is possible based on the membership functions of the concepts. When the amount of information is large it is very important to be able to give the most relevant, or most probably relevant search result in the beginning of the result list. As will be shown in the evaluation of Chapter 13, *FFSS*'s ranking performance is good.

Enabling the separation of end-user views from annotation ontologies In many cases the formal ontologies created by and for domain experts are not ideal for the end-user to search. The concepts are not familiar to a non-expert and the organization of the ontology may be unintuitive. When search facets are algorithmically projected from these annotation ontologies, as described in Chapter 3, the resulting search *GUI* might end-up being rather complicated and unintuitive. This sometimes undermines the main goal of faceted search systems, i.e., providing intuitive and easy user interfaces for exploratory search. In this paper we tackled the problem by creating a way to represent search views separately from the ontologies and to map the search concepts to the annotation concepts. The mappings may contain uncertainty.

The methods presented in this part of the dissertation were later utilized in the *HealthFinland* portal²⁰, which is a national Finnish health portal for the common citizen [60].

²⁰<http://www.terveysuomi.fi>

8.2 Lessons Learned

Both the *FFSS* paradigm, and the ontological extension of the *TF-IDF* weighting method proved to be rather straight forward to design and implement. *CFSS* can be seen as a special case of the fuzzy framework such that the annotations and the mappings have the weight 1.0, i.e. are crisp. According to the case study that will be presented in Chapter 13, the framework provides good ranking results.

Our framework did get some inspiration from the fuzzy versions of description logics. We share the idea of generalizing the set theoretic basis of an IR-system to fuzzy sets in order to enable the handling of vagueness and uncertainty. In addition, the use of fuzzy implication to reason about fuzzy inclusion between concepts is introduced in the fuzzy version [97] of the description logic *SHOIN(D)*. However, the ontologies that we use are mainly simple concept taxonomies, and in many practical cases we saw it as an unnecessary overhead to anchor our framework in description logics.

In terms of computation complexity, *FFSS* is feasible, because it adds only complexity of ordering the search items according to relevance *CFSS* paradigm, which has proved to be a rather scalable framework. For example, the *FSS* engine *OntoViews* was tested to scale up to 2.3 million search items and 275,000 search categories in [72].

However, the fuzzy logic approach is criticized because of the arbitrariness in finding the numeric values needed and lack of solid mathematical basis [98]. A concrete deficiency related to this is that *FFSS* does not provide mathematically well-founded ways to combine evidence of search item relevance from multiple sources. For example, documents could be ranked based on relevance feedback given by the users, by standard *TF-IDF*-based relevance measures, or some other heuristics. We would like to be able to combine the relevance measures given by different ranking algorithms in a sound way, because it has been shown that the combination of evidence

from multiple sources improves ranking [70]. For example, we would like to be able to quantify our level of trust in each ranking algorithm so that this trust will be taken into account in the final ranking. Naturally we could combine rankings e.g. by fuzzy union or intersection, however, this would be just a heuristic combination. On the other hand, probability theory has been shown to provide mathematically well-founded mechanisms for evidence combination [20]. As a result of these considerations we decided to develop also probabilistic solutions to the problems of *FSS*. The probabilistic approach will be presented next.

Part III

PROBABILISTIC APPROACH

9 Introduction to the Probabilistic Approach

This part of the dissertation will present probabilistic solutions to the problems of *CFSS* presented in Chapter 4, and it also provides a flexible way to combine evidence of relevance from multiple sources. The problems and solution are outlined in Table 9.1.

Table 9.1: The problems and solutions presented in this part of the dissertation.

	Problem	Solution	Chapter
1.	Semantic Web ontologies lack the support for modeling uncertainty inherent in the world, including the <i>FSS</i> system itself.	A graph notation for representing uncertainty and conceptual overlap in Semantic Web taxonomies, and a Bayesian method for computing degrees of overlap between the concepts of such a taxonomy.	10.
2.	Crisp faceted semantic search lacks the capability to rank search results based on relevance.	Ranking of documents based on probability of relevance inferred based on weighted annotations, ontology structure, and mappings of end-user search facets to annotation ontologies.	11.
3.	Concepts of annotation ontologies are not always suitable to be presented as search categories on the search GUI.	Search facets are defined separately and probabilistically mapped to annotation ontologies. Boolean combinations of annotation concepts can be used in the mappings.	11.
4.	How to combine rankings of search result provided by different schemes to provide better ranking of search results?	Rankings of multiple ranking schemes are probabilistically combined to reach the final probability of relevance of the document. A number of ranking schemes are developed.	10., 11.

The lack of support of Semantic Web ontologies for uncertainty modeling is tackled by creating a method to probabilistically exploit the concept hierarchies which usually constitute the backbone of Semantic Web ontologies. Ranking of search results is supported by the developed *probabilistic faceted semantic search (PFSS)* framework, based on weighted annotations, probabilistically interpreted concept hierarchies found in ontologies, and mappings between search facets and annotation ontologies. To solve the third problem, in *PFSS* search facets are separately defined

and mapped to annotation ontologies. As in *FFSS*, also in *PFSS* Boolean combinations of annotation concepts can be used in the mappings. The problem of evidence combination is solved by creating a probabilistic mechanism to combine evidence from multiple ranking schemes.

The solutions presented in this part are based on Bayesian probability, which is a branch of probability theory, where probability is interpreted as a degree of belief. This is plausible, because it has been shown, that an agent's rational degrees of belief follow the rules of probability. More specifically, if one wishes to represent the *plausibility* of a proposition by a real number and requires consistency in the resulting calculus, the axioms of probability follow logically [111, 29]. Thus, Bayesian probability theory is more than a heuristic for modeling uncertainty, but a formalism that truly captures rational inference under uncertainty. This also means, that fuzzy logic, which has a different set of axioms, necessarily violates the consistency requirements [26].

Intuitively, it is easy to grasp how semantic search could be modeled using Bayesian probability. One can view the search engine as uncertain about the information needs of the user [24]. This uncertainty can be reduced by utilizing the information contained in the semantic knowledge base, which the search application is based on. The system then makes inferences about the probabilities of relevance of search items, and based on these inferences it decides the ranking of search results. Evidence combination is at the core of the Bayesian probability theory, and thus it seems a good approach for combining evidence of document relevance, which has been shown to improve quality of ranking [70] of search systems.

There are, of course, a lot of modeling issues. For example, the semantic knowledge base can be interpreted probabilistically in more than one way, and we will see that there are already numerous probabilistic methods to model uncertainty related to Semantic Web ontologies [85, 32, 41, 79, 63], each of which has usually been

developed with one application area in mind. Also the information retrieval event itself can be modeled in different ways, and as a consequence, there are also numerous probability models developed for information retrieval [88, 20, 104, 91, 24].

The first probabilistic information retrieval method was the *Probabilistic Ranking Principle Approach* [88], which was an extension to standard Boolean information retrieval, such that the index terms of documents were weighted based on the importance of the index term to the document. Later, also other probabilistic methods have been developed, such as the *Inference Network Model for IR* [104], and the *Belief Network Model* [91], both of which model the information retrieval system as a Bayesian network. Both these methods still adhere to the *bag of words* principle, in which the documents are seen as bags of index terms, and those index terms themselves are seen as mutually independent entities. However, the Bayesian network based approaches above have already shown, that probabilistic information retrieval approaches are suitable to combination of evidence to improve rankings. *Latent Dirichlet Allocation* [24], has a different approach, in the sense that it models each document as a probability distribution over topics, and each topic is represented by a probability distribution of words. Thus, the model has a hierarchical structure. However, this model is constructed bottom-up by analyzing the words in a document collection. Despite of the amount of work done, it seems that a complete framework for probabilistic semantic search and probabilistic faceted semantic search, in particular, is still missing, and this part of the dissertation will present such a framework.

One potential problem of such a probabilistic semantic search system is computational complexity. Even though Bayesian networks [83] have improved the applicability of probability theory to a great degree, a Bayesian network modeling the ontological concepts, documents, and search facets in a faceted semantic search application might still prove to be too inefficient for real world use. However, from the point of view of computational complexity, faceted search has the virtue of prepar-

ing the way for precomputation. In fact, all search category specific probabilities of document relevance can be precomputed. The only remaining part for on-line computation is the probability of document relevance for the multi-faceted selection based on the individual search category specific probabilities.

The rest of this part of the dissertation is organized as follows: First, a short preliminary of Bayesian probability and Bayesian networks needed to follow the development of the probabilistic solutions is given. Then, in Chapter 10 a method for modeling uncertainty in Semantic Web taxonomies will be presented, which will also include comparison to related methods found in the literature. Chapter 11 will develop the *PFSS* framework, and a reference implementation of it is presented in Appendix A. The contributions of—and lessons learned from—the work presented in this part, are then summarized in Chapter 12. An empirical evaluation of *PFSS* and *FFSS* will be presented in the final part of this dissertation, in Chapter 13.

9.1 Basic Probability Theory

Probability theory is concerned with experiments that have outcomes. The set of all outcomes is called the *sample space*, and is denoted Ω . For example, for a minimalistic faceted search system composed of two documents $D1$ and $D2$, the set of outcomes—i.e., the *elements* of Ω —is:

- $D1$ is relevant, $D2$ is relevant
- $D1$ is relevant, $D2$ is not relevant
- $D1$ is not relevant, $D2$ is relevant
- $D1$ is not relevant, $D2$ is not relevant

A subset of the sample space is called an *event*, and a subset containing exactly one element is called an *elementary event*.

A function that assigns a real number $P(E)$ to each event $E \subseteq \Omega$, where $\Omega = \{e_1, e_2, \dots, e_n\}$ is called a *probability function* on the set of subsets of Ω if it satisfies the following conditions:

1. $0 \leq P(\{e_i\}) \leq 1$ for $1 \leq i \leq n$.
2. $P(\{e_1\}) + P(\{e_2\}) + \dots + P(\{e_n\}) = 1$.
3. For each event $E = \{e_{i_1}, e_{i_2}, \dots, e_{i_k}\}$ that is not an elementary event, $P(E) = P(\{e_{i_1}\}) + P(\{e_{i_2}\}) + \dots + P(\{e_{i_k}\})$.

If P is a probability function, then $P(E)$ is called the *probability of E* . The pair (Ω, P) is called a probability space. According to the principle of indifference, elementary events are to be considered equiprobable if we have no reason to expect or prefer one over the other. According to this principle when there are n elementary events, the probability of each is the ratio $1/n$.

If E and F are events, s.t. $F \neq \emptyset$, then $P(E|F)$ is the *conditional probability of event E given the event F* . It given by:

$$P(E|F) = P(E \cap F)/P(F). \quad (9.1)$$

The intuition comes from probability as a ratio, i.e., the fraction of outcomes in E that are also outcomes in F , i.e.

$$P(E \cap F)/P(F) = \frac{n_{EF}/n}{n_F/n} = \frac{n_{EF}}{n_F}. \quad (9.2)$$

where n is the number of outcomes in a sample space, n_{EF} is the number of outcomes in $E \cap F$, and n_F the number of outcomes in F .

Given a probability space (P, Ω) , a random variable X is a function over Ω , s.t. it assigns a value of X to each element of Ω . X^j denotes all the elements in Ω that the function X maps to the value j of X . $X^j \subseteq \Omega$, so each value of a variable is an event. A random variable induces a probability function over the set of its values (the space of X). This probability function is usually called the probability distribution of X .

For example, in our minimalistic faceted search system we could define the binary random variable D_1 such that value D_1^1 represents the outcomes

- D_1 is relevant, D_2 is relevant
- D_1 is relevant, D_2 is not relevant

and D_1^0 represents the outcomes

- D_1 is not relevant, D_2 is relevant
- D_1 is not relevant, D_2 is not relevant

Thus, D_1 is a binary random variable that represents the relevance of document D_1 . Similarly, we could define the random variable D_2 to represent the relevance of document D_2 . If our system contained a search category SC_1 the result set of which contained D_1 but not D_2 then we could define a binary random variable SC_1 such that SC_1^1 represents the outcome *D_1 is relevant, D_2 is not relevant*, and SC_1^0 represents all the other three outcomes in Ω .

Now, the conditional probability of D_1 being relevant given that SC_1 is selected is:

$$P(D_1^1|SC_1^1) = n_{D_1^1SC_1^1}/n_{SC_1^1} = 1/1 = 1, \quad (9.3)$$

And the conditional probability that D_2 is relevant given SC_1 is:

$$P(D_2^1|SC_1^1) = n_{D_2^1SC_1^1}/n_{SC_1^1} = 0/1 = 0. \quad (9.4)$$

Two random variables X and Y , s.t., $P(X) \neq 0$, and $P(Y) \neq 0$, are independent if for all values of X and Y

$$P(X^i|Y^j) = P(X^i) \quad (9.5)$$

For example, it can be easily seen that D_1 and D_2 above are independent. However, SC_1 and D_1 are not independent, because $P(D_1^1|SC_1^1) = 1 \neq P(D_1^1)$.

Two or more random variables induce a probability function over the cartesian product of their spaces. This multivariable distribution is called the joint probability distribution of the variables.

For example the joint probability distribution of D_1 and SC_1 is:

- $P(D_1^1, SC_1^1) = P(D_1^1 \cap SC_1^1) = n_{D_1^1SC_1^1}/n = 1/4$,
- $P(D_1^1, SC_1^0) = P(D_1^1 \cap SC_1^0) = n_{D_1^1SC_1^0}/n = 1/4$,
- $P(D_1^0, SC_1^1) = P(D_1^0 \cap SC_1^1) = n_{D_1^0SC_1^1}/n = 0/4$,
- $P(D_1^0, SC_1^0) = P(D_1^0 \cap SC_1^0) = n_{D_1^0SC_1^0}/n = 2/4$

As can be seen this distribution is a probability function.

9.2 Bayesian Networks

A Bayesian network is a graphical formalism that offers a natural way of representing the probabilistic independencies satisfied by a probability function, and for this

reason it can be used to efficiently represent a probability function [111]. Efficient inference algorithms, that take advantage of the probabilistic independencies among the variables, have been developed for Bayesian networks. A Bayesian network consists of two components:

1. A directed acyclic graph (DAG), where a set of random variables makes up the nodes of the network and a set of directed links connects pairs of nodes. An example DAG is presented in Figure 9.1. For any node X in a DAG, the nodes from which there are links pointing to X , are called the *parents* of X , often denoted Par_X . The links originating from X point to nodes that are called the *children* of X , often denoted Chi_X . The *ancestors* of X are its parents, the parents of the parents and so on. The *descendants* of X are its children, the children of the children and so on. A node that does not have any parents is a *root* node. In the DAG of Figure 9.1, for example, the parents of C are A and D , and the ancestors of C are A , B , and D . C does not have any children or descendants. The children of B are A and D , and its descendants are A , D , C , and E . B does not have any parents, so it is a root node. Semantically, a link from one node to another means that the first has a direct influence on the second.

2. A Probability specification, which specifies the probability distributions of each variable X in the DAG, conditional on its parents. This specification is usually given by a conditional probability table, and it quantifies the influence of the parent variables on X . The conditional probability table contains a cell for each assignment to X conditional on each assignment to the parents of X . As an example, Table 9.2 contains a possible conditional probability table for the variable C of Figure 9.1 under the assumption that all the variables in the network are binary random variables, i.e., have two possible assignments superscripted by 0 and 1.

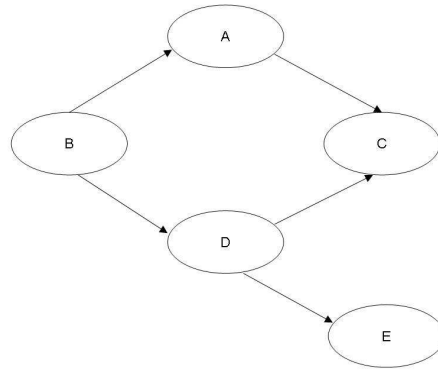


Figure 9.1: An example of a DAG.

Table 9.2: An example conditional probability table for variable C of DAG of Figure 9.1.

$P(C^0 A^0D^0) = 0.3$	$P(C^1 A^0D^0) = 0.7$
$P(C^0 A^0D^1) = 0.8$	$P(C^1 A^0D^1) = 0.2$
$P(C^0 A^1D^0) = 0.5$	$P(C^1 A^1D^0) = 0.5$
$P(C^0 A^1D^1) = 0.1$	$P(C^1 A^1D^1) = 0.9$

A fundamental feature of a Bayesian network is stated by the *Markov Condition*, which says that conditional on its parents any variable is probabilistically independent of all other variables apart from its descendants. This is the reason for the efficiency of the Bayesian network representation of a joint probability distribution. Under normal circumstances the joint probability distribution of a set of variables is specified by the chain rule. For example, for the variables of the DAG of Figure 9.1, the joint probability distribution computed by the chain rule would be $P(A, B, C, D, E) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)P(E|A, B, C, D)$. The size of this representation, i.e., the number of parameters (individual probabilities) required by the probability specification, is the number of assignments in the joint distribution minus one, in our example this is $2^5 - 1 = 31$.

Thanks to the structure of the Bayesian network, it can be represented in the more efficient form $P(A, B, C, D, E) = P(B)P(A|B)P(D|B)P(C|A, D)P(E|D)$, where the size of the representation is the sum of the sizes of the representations of each variable's distributions. The size of a variable's distributions is the number of assignments to the variable minus 1, multiplied by the number of assignments to its parents (which is 1 in the case of a root variable). In our example the size of the Bayesian network representation is thus $1 + 2 + 2 + 4 + 2 = 11$ [111]. This factorization of the joint probability distribution enables the use of efficient reasoning algorithms. As applied to Bayesian networks, reasoning means computing the probability distribution of values of the variable Y when we know the value (or probability distribution of values) of X .

10 Modeling Degrees of Overlap Between Concepts in Semantic Web Taxonomies

This chapter is largely based on the article:

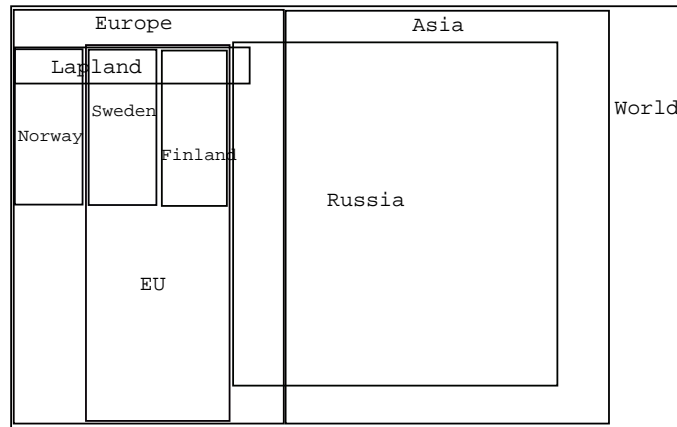
Markus Holi and Eero Hyvönen. 2006. Modeling Uncertainty in Semantic Web Taxonomies[47]. In: Soft Computing in Ontologies and Semantic Web (Zhongmin Ma (ed.)), Springer-Verlag.

This section provides the probabilistic extension to ontologies which will be used in the probabilistic faceted semantic search framework presented in the next chapter.

10.1 The Problem and the Solution Approach

The Venn diagram of Figure 10.1 illustrates some countries and areas in the world. A crisp *partOf* meronymy cannot represent the partial overlap between the geographical area Lapland and the countries Finland, Sweden, Norway, and Russia, for example. A frequently used way to model the above situation would be to represent Lapland as the direct meronym of all the countries it overlaps, as in Figure 10.2. This structure, however does not represent the situation of the map correctly, because Lapland is not subsumed by anyone of these countries. In addition, the transitivity of the subsumption relation disappears in this structure. See, for example, the relationship between Lapland and Asia. In the Venn diagram they are disjoint, but according to the taxonomy, Lapland is subsumed by Asia.

Another way would be to partition Lapland according to the countries it overlaps, as in Figure 10.3. Every part is a direct meronym of both the respective country and Lapland. This structure is correct, in principle, but it too does not contain enough



World $37 \times 23 = 851$
 Europe $15 \times 23 = 345$
 Asia $18 \times 23 = 414$
 EU $8 \times 21 = 168$
 Sweden $4 \times 9 = 36$
 Finland $4 \times 9 = 36$
 Norway $4 \times 9 = 36$
 Lapland $13 \times 2 = 26$ Lapland & (Finland | Sweden | Norway) = 8
 Lapland & EU = 16 Lapland & Russia = 2
 Russia $18 \times 19 = 342$ Russia & Europe = 57 Russia & Asia = 285

Figure 10.1: A Venn diagram illustrating countries, areas, their overlap, and size in the world.

information to make inferences about the degrees of overlap between the areas. It does not say anything about the sizes of the different parts of Lapland, and how much they cover of the whole area of Lapland and the respective countries.

According to Figure 10.1, the size of Lapland is 26 units, and the size of Finland is 36 units. The size of the overlapping area between Finland and Lapland is 8 units. Thus, $8/26$ of Lapland belongs to Finland, and $8/36$ of Finland belongs to Lapland. On the other hand, Lapland and Asia do not have any overlapping area, thus no part (0) of Lapland is part of Asia, and no part of Asia is part of Lapland. If we want a taxonomy to be an accurate representation of the 'map' of Figure 10.1, there should be a way to make this kind of inferences based on the taxonomy.

Our method enables the representation of overlap in taxonomies, and the compu-

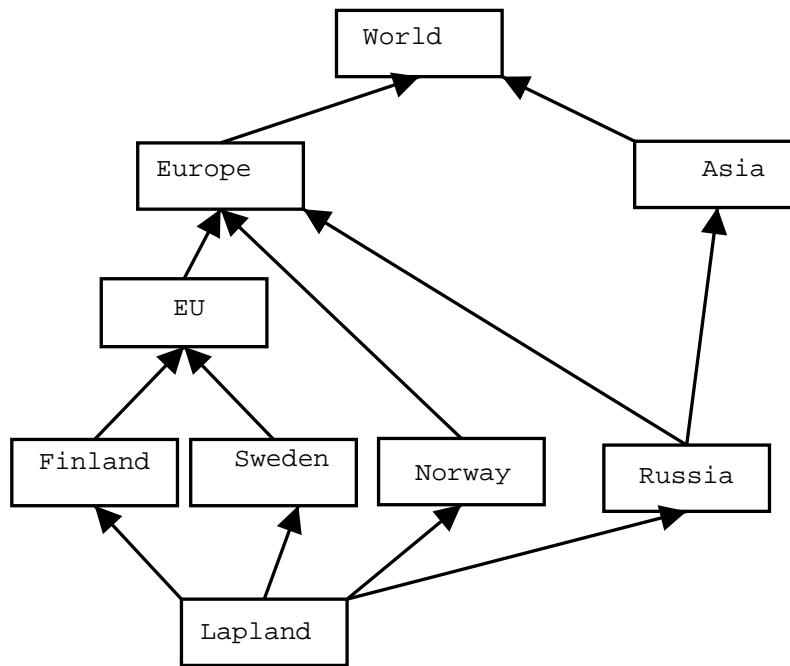


Figure 10.2: A standard Semantic Web taxonomy based on the Venn diagram of Figure 10.1.

tation of overlap between a *selected* concept and every other, i.e. *referred* concept in the taxonomy. Thus, an *overlap table* is created for the selected concept. The overlap table can be created for every concept of a taxonomy. For example, Table 10.1 presents the overlap table of Lapland based on the the Venn diagram of Figure 10.1. The Overlap column lists values expressing the mutual overlap of the selected concept and the other - referred - concepts, i.e., $Overlap = \frac{|Selected \cap Referred|}{|Referred|} \in [0, 1]$.

Intuitively, the overlap value has the following meaning: The value is 0 for disjoint concepts (e.g., Lapland and Asia) and 1, if the referred concept is subsumed by the selected one. High values lesser than one imply, that the meaning of the selected concept approaches the meaning of the referred one.

This overlap value can be used in information retrieval tasks. Assume that an ontology contains individual products manufactured in the different countries and areas

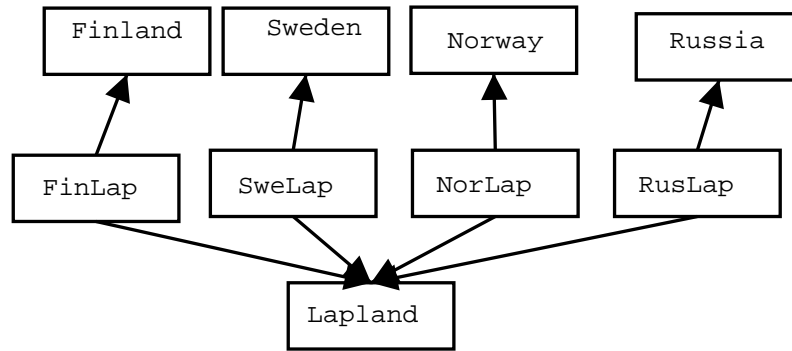


Figure 10.3: Representing Lapland’s overlaps by partitioning it according to the areas it overlaps. Each part is subsumed by both Lapland and the respective country.

of Figure 10.1. The user is interested in finding objects manufactured in Lapland. The overlap values of Table 10.1 then tell how well the annotations “Finland”, “EU”, “Asia”, etc., match with the query concept “Lapland” in a well-defined probabilistic sense, and the hit list can be sorted into an order of relevance accordingly.

The overlap value between the selected concept (e.g. Lapland) and the referred concept (e.g. Finland) can in fact be written as the conditional probability $P(\text{Finland}'|\text{Lapland}')$ whose interpretation is the following: If a person is interested in data records about Lapland, what is the probability that the annotation “Finland” matches her query? X' is a binary random variable such that $X' = \text{true}$ means that the annotation “X” matches the query, and $X' = \text{false}$ means that “X” is not a match. This conditional probability interpretation of overlap values will be used in Section 10.4 where the computation of the overlaps is explained. Notice that the modeling of overlap between geographical concepts, as in our example, is truly uncertain, because the exact amount of overlap is never known.

It is mathematically easy to compute the overlap tables, if a Venn diagram (the sets) is known. In practice, the Venn diagram may be difficult to create from the modeling view point, and computing with explicit sets is computationally complicated and inefficient. For these reasons our method calculates the overlap values

Table 10.1: The *overlap table* of Lapland according to Figure 10.1.

Selected	Referred	Overlap
Lapland	World	$26/851 = 0.0306$
	Europe	$26/345 = 0.0754$
	Asia	$0/414 = 0.0$
	EU	$16/168 = 0.0953$
	Norway	$8/36 = 0.2222$
	Sweden	$8/36 = 0.2222$
	Finland	$8/36 = 0.2222$
	Russia	$2/342 = 0.0059$

from a taxonomic representation of the Venn diagram.

Our method consists of two parts:

1. A graphical notation by which partial subsumption and concepts can be represented in a quantified form. The notation can be represented easily in RDF.
2. The computation of the degrees of overlap between the concepts of a taxonomy. We will present two methods to compute the overlap: The first method is a tailored algorithm based directly on the taxonomical structure, and the second method quantifies the overlap by transforming the taxonomy first into a Bayesian network [36].

10.2 Representing Overlap

In RDFS and OWL a concept, often represented by a class refers to a set of individuals. Subsumption reduces essentially into the subset relationship between the sets corresponding to the classes. A taxonomy is therefore a set of sets and can be represented, e.g., by a Venn diagram.

If A and B are sets, then A must be in one of the following relationships to B .

1. A is a subset of B ; i.e., $A \subseteq B$.
2. A partially overlaps B ; i.e., $\exists x, y : (x \in A \wedge x \in B) \wedge (y \in A \wedge y \notin B)$.
3. A is disjoint from B ; i.e., $A \cap B = \emptyset$.

Based on these relations, we have developed a simple graph notation for representing uncertainty and overlap in a taxonomy as an acyclic *overlap graph* (*OG*). Here concepts are nodes, and a number called *mass* is attached to each node. The mass of the concept A is a measure of the size of the set corresponding to A , i.e. $m(A) = |s(A)|$, where $s(A)$ is the set corresponding to A . A solid directed arc from concept A to B denotes crisp subsumption $s(A) \subseteq s(B)$, a dashed arrow denotes disjointness $s(A) \cap s(B) = \emptyset$, and a dotted arrow represents quantified partial subsumption between concepts, which means that the concepts partially overlap in the Venn diagram. The amount of overlap is represented by the *partial overlap value* $p = \frac{|s(A) \cap s(B)|}{|s(A)|}$.

In addition to the quantities attached to the dotted arrows, also the other arrow types have implicit overlap values. The overlap value of a solid arc is 1 (crisp subsumption) and the value of a dashed arc is 0 (disjointness). The quantities of the arcs emerging from a concept must sum up to 1. This means that either

only one solid arc can emerge from a node or several dotted arcs (partial overlap). In both cases, additional dashed arcs can be used (disjointness). Intuitively, the outgoing arcs constitute a quantified partition of the concept. Thus, the dotted arrows emerging from a concept must always point to concepts that are mutually disjoint with each other.

Notice that if two concepts overlap, there must be a directed (solid or dotted) path between them. If the path includes dotted arrows, then (possible) disjointness between the concepts must be expressed explicitly using the disjointness relation. If the directed path is solid, then the concepts necessarily overlap.

For example, Figure 10.4 depicts the meronymy of Figure 10.1 as an overlap graph. The geographic sizes of the areas are used as masses and the partial overlap values are determined based on the Venn diagram. This graph notation is complete in the sense that any Venn diagram can be represented by it [45]. However, sometimes the accurate representation of a Venn diagram requires the use of auxiliary concepts, which represent results of set operations over named sets, for example $s(A) \setminus s(B)$, where A and B are ordinary concepts and \setminus denotes the subtraction operation between sets.

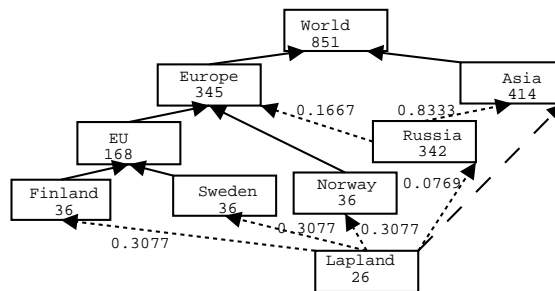


Figure 10.4: The taxonomy corresponding to the Venn diagram of Figure 10.1.

10.3 Solid Path Structure

Our method creates an overlap table (as in Figure 10.1) for each concept in the taxonomy. Computing the overlaps is easiest when there are only solid arcs; i.e., complete subsumption relation, between concepts. If there is a directed solid path from A (selected) to B (referred), then overlap $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{m(A)}{m(B)}$. If the solid path is directed from B to A , then $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{m(B)}{m(B)} = 1$. If there is not a directed path between A and B , then $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{|\emptyset|}{m(B)} = 0$.

If there is a mixed path of solid and dotted arcs between A and B , then the calculation is not as simple. Consider, for example, the relation between *Lapland* and *EU* in Figure 10.4. To compute the overlap, we have to follow all the paths emerging from *Lapland*, take into account the disjoint relation between *Lapland* and *Asia*, and sum up the partial subsumption values somehow.

To exploit the simple solid arc case, a taxonomy with partial overlaps is first transformed into a *solid path structure*, in which crisp subsumption is the only relation between the concepts. The transformation is done by using to the following principle:

Transformation Principle 1 *Let A be the direct partial subconcept of B with overlap value o . In the solid path structure the partial subsumption is replaced by an additional middle concept, that represents $s(A) \cap s(B)$. It is marked to be the complete subconcept of both A and B , and its mass is $o \cdot m(A)$.*

For example, the taxonomy of Figure 10.4 is transformed into the solid path structure of Figure 10.5. The original partial overlaps of *Lapland* and *Russia* are transformed into crisp subsumption by using middle concepts.

```

Data: OverlapGraph T
Result: SolidPathStructure SPS
1 SPS := empty;
2 foreach concept c in T do
3   foreach complete or partial direct superconcept p of c in T do
4     if p connected to its superconcepts through middle concepts in SPS then
5       mc := the middle concept that c overlaps;
6       if c complete subconcept of p then
7         mark c to be complete subconcept of mc in SPS;
8       else
9         newMc := middle concept representing
10         $s(c) \cap s(p)$ ;
11        mark newMc to be complete subconcept of c and mc in SPS;
12      end
13    else
14      if c complete subconcept of p then
15        mark c as complete subconcept of p in SPS;
16      else
17        newMc := middle concept representing
18         $s(c) \cap s(p)$ ;
19        mark newMc to be complete subconcept of c and p in SPS;
20      end
21    end
22  end
23 end

```

Algorithm 3: Creating the solid path structure

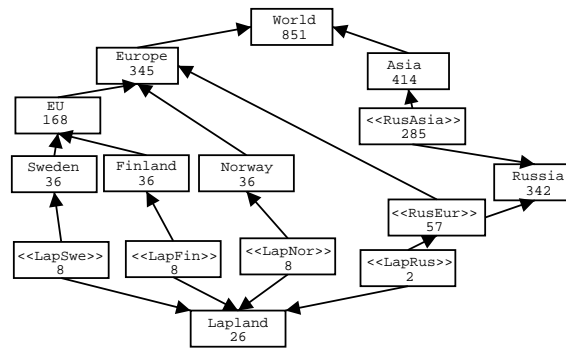


Figure 10.5: The taxonomy of Figure 10.4 as a solid path structure.

The transformation is specified in Algorithm 3. The algorithm processes the overlap graph T in a breadth-first manner starting from the root concept. A concept c is processed only after all of its super concepts (partial or complete) are processed. Because the graph is acyclic, all the concept will eventually be processed.

Each processed concept c is written to the solid path structure SPS . Then each arrow emerging from c is processed in the following way. If the arrow is solid, indicating subsumption, then it is written into the solid path structure as such. If the arrow is dotted, indicating partial subsumption, then a middle concept $newMc$ is added into the solid path structure. It is marked to be the complete subconcept of both c and the concept p to which the dotted arrow points in T . The mass of $newMc$ is $m(newMc) = |s(c) \cap s(p)| = o \cdot m(c)$, where o is the overlap value attached to the dotted arrow.

However, if p is connected to its superconcepts (partial or complete) with a middle concept structure, then the processing is not as simple. In that case c has to be connected to one of those middle concepts. The right middle concept is found by using the information conveyed in the dashed arcs emerging from c . The right middle concept mc is the one that is not subsumed by a concept that is marked to be disjoint from c in the overlap graph. This is the middle concept that c overlaps.

Notice, that if the overlap graph is an accurate representation of the underlying Venn diagram, then mc is the only middle concept that fulfills the condition.

If c is a complete subconcept of p in the overlap graph T , then c is marked to be the complete subconcept of mc in SPS . If c is a partial subconcept of p in T , then it is connected to mc with a middle concept structure.

Notice, that if c was connected directly to p , instead of mc , then the information conveyed in the dashed arrows, indicating disjointness between concepts would have been lost. For example, if in Figure 10.5 *Lapland* was connected directly to *Russia*, then the information about the disjointness of *Lapland* and *Asia* would have been lost.

10.4 Computing the Overlaps

We have implemented two alternatives for computing the overlaps. The first is based directly on the solid path structure, and the second transforms the solid path structure into a Bayesian network which can be used to compute the conceptual overlap. Both alternatives are presented here.

10.4.1 The Solid Path Structure Approach

The overlap table values o for a selected concept A and a referred concept B can be calculated from the solid path structure using Equation 10.1, where C denotes the set of common subconcepts of A and B , such that there are not two concepts $C_j, C_k \in C, j \neq k$, such that C_j subsumes C_k . The overlap table for A can be computed by going through all the concepts of the graph and calculating the overlap value according to the equation.

$$o = \begin{cases} 1 & \text{if } A \text{ subsumes } B, \\ 0 & \text{if } C = \emptyset, \\ \frac{\sum_{c \in C} m(c)}{m(B)} & \text{otherwise} \end{cases} \quad (10.1)$$

Equation 10.1 is composed of three cases. The first case states that if B is a subconcept of A then the overlap value o is always 1. This is because then B is completely covered by A . The second case states that if the set C is empty then o is 0. If this set is empty, then A and B are disjoint, and the overlap value o is 0. The third case states that if C is not empty, then o equals the sum of the masses of the concepts belonging to C divided by the mass of B .

Recall from Section 10.1, that the overlap value between A (selected) and B (referred) can be interpreted as the conditional probability $P(B'|A')$, where X' is a binary random variable such that $X' = true$ or X'^1 represents the situation that the user is interested in the concept X , and $X' = false$ or X'^0 represents the situation that the user is not interested in that concept. We will use this interpretation of the overlap value in the next chapter (11) when developing the probabilistic faceted semantic search (*PFSS*) framework.

Notice that the Venn diagram from which $s(A)$ and $s(B)$ are taken is not interpreted as a probability space, and the elements of the sets are not interpreted as elementary outcomes of some random phenomenon. The overlap value between $s(A)$ and $s(B)$ is used merely as a means for determining the conditional probability defined above.

10.4.2 The Bayesian Network Approach

Because the overlap values between concepts can be interpreted as conditional probabilities, we chose to implement the computation of overlap values also by trans-

forming the solid path structure into a Bayesian network. This alternative approach was created because we think that the ability to represent the model as a Bayesian network proves that it is probabilistically sound.

In this approach we use the solid path structure as a Bayesian network topology. In the Bayesian network the Boolean random variable X' replaces the concept X of the solid path structure.

The joint probability distribution of the Bayesian network is defined by conditional probability tables (CPT) $P(A'|B'_1, B'_2, \dots, B'_n)$ for nodes with parents $B'_i, i = 1 \dots n$, and by prior marginal probabilities set for nodes without parents. The CPT $P(A'|B'_1, B'_2, \dots, B'_n)$ for a node A' can be constructed by enumerating the value combinations (true/false) of the parents $B'_i, i = 1 \dots n$, and by assigning:

$$P(A' = true|B'_1 = b_1, \dots, B'_n = b_n) = \frac{\sum_{i \in \{i: b_i = true\}} m(B_i)}{m(A)} \quad (10.2)$$

The value for the complementary case $P(A' = false|B'_1 = b_1, \dots, B'_n = b_n)$ is obtained simply by subtracting from 1. The above formula is based on Equation 10.1. The intuition behind the formula is the following. If a user is interested in Sweden and in Finland, then she is interested both in data records about Finland and in data records about Sweden. The set corresponding to this is $s(Finland) \cup s(Sweden)$. In terms of the *OG* this is written as $m(Finland) + m(Sweden)$. In the Bayesian network both Finland and Sweden will be set “true”. Thus, the bigger the number of European countries that the user is interested in, the bigger the probability that the annotation “Europe” matches her query, i.e., $P(Europe' = true|Sweden' = true, Finland' = true) > P(Europe' = true|Finland' = true)$. As an example, Table 10.2 presents the complete CPT for the variable EU' along with a verbal interpretation of each case. As can be observed, each conditional probability has a well

defined probabilistic meaning, however, as we will see in Chapter 11, in the context of probabilistic faceted search we will not explicitly ask for conditional probabilities with negative given values.

If A' has no parents, then $P(A' = true) = \lambda$, where λ is a very small non-zero probability, because we want the posterior probabilities to result from conditional probabilities only, i.e., from the overlap information. The whole overlap table of a concept can now be determined efficiently by using the Bayesian network with its conditional and prior probabilities. By instantiating the nodes corresponding to the selected concept and the concepts subsumed by it as evidence (their values are set “true”), the propagation algorithm returns the overlap values as posterior probabilities of nodes.

Notice that when using the Bayesian network in the above way, a small inaccuracy is attached to each value as the result of the λ prior probability that was given to the parentless variables. This error approaches zero as λ approaches zero. Despite this small inaccuracy we decided to define the Bayesian network in the above manner for the following reasons.

First, to be able to easily use the the solid path structure as the topology of the Bayesian network. The CPTs can be calculated directly based on the masses of the concepts. Second, with this definition the Bayesian evidence propagation algorithm returns the overlap values readily as posterior probabilities. We experimented with various ways to construct a Bayesian network according to probabilistic interpretations of the Venn diagram. However, none of these constructions answered to our needs as well as the construction described above.

Table 10.2: The conditional probability table for the random variable EU' along with the verbal interpretation of each case.

$P(EU'^1 Sweden'^1, Finland'^1) = (m(Sweden) + m(Finland)) / m(EU) = 72 / 168 = 0.429$ <p>The probability that the user is interested in EU if the user is interested in both Finland and Sweden.</p>
$P(EU'^0 Sweden'^1, Finland'^1) = 1 - P(EU'^1 Sweden'^1, Finland'^1) = 1 - 0.429 = 0.571$ <p>The probability that the user is not interested in EU if the user is interested in both Finland and Sweden.</p>
$P(EU'^1 Sweden'^1, Finland'^0) = m(Sweden) / m(EU) = 36 / 168 = 0.214$ <p>The probability that the user is interested in EU if the user is interested in Sweden but not in Finland.</p>
$P(EU'^0 Sweden'^1, Finland'^0) = 1 - P(EU'^1 Sweden'^1, Finland'^0) = 1 - 0.214 = 0.786$ <p>The probability that the user is not interested in EU if the user is interested in Sweden but not in Finland.</p>
$P(EU'^1 Sweden'^0, Finland'^1) = m(Finland) / m(EU) = 36 / 168 = 0.214$ <p>The probability that the user is interested in EU if the user is interested in Finland but not in Sweden.</p>
$P(EU'^0 Sweden'^0, Finland'^1) = 1 - P(EU'^1 Sweden'^0, Finland'^1) = 1 - 0.214 = 0.786$ <p>The probability that the user is not interested in EU if the user is interested in Finland but not in Sweden.</p>
$P(EU'^1 Sweden'^0, Finland'^0) = 0 / m(EU) = 0 / 168 = 0$ <p>The probability that the user is interested in EU if the user is interested neither in Finland nor in Sweden.</p>
$P(EU'^0 Sweden'^0, Finland'^0) = 1 - P(EU'^1 Sweden'^0, Finland'^0) = 1 - 0 = 1$ <p>The probability that the user is not interested in EU if the user is interested neither in Finland nor in Sweden.</p>

10.5 Implementation

The presented method has been implemented as a proof-of-concept. The implementation uses the Bayesian network approach for computing the overlaps.

10.5.1 Overlap Graph

To comply with Semantic Web representation standards, overlap graphs are represented as RDF ontologies in the following way. Concepts are represented as RDFS classes²¹. The concept masses are represented using a special *Mass* class. It has two properties, subject and mass that tell the concept resource in question and the mass as a numeric value, respectively. The subsumption relation can be implemented with a property of the user's choice. Partial subsumption is implemented by a special *PartialSubsumption* class with three properties: subject, object and overlap. The subject property points to the direct partial subclass, the object to the direct partial superclass, and overlap is the partial overlap value. The disjointness arc is implemented by the *disjointFrom* property used in OWL.

10.5.2 Overlap Computations

The architecture of the implementation can be seen in Figure 10.6. The input of the implementation is an RDF ontology, the URI of the root node of the overlap graph, and the URI of the subsumption property used in the ontology. Additionally, also an RDF data file that contains data records annotated according to the ontology may be given. The output is the overlap tables for every concept in the taxonomy extracted from the input RDF ontology. Next, each submodule in the system is discussed briefly.

²¹Actually, any resources including instances could be used to represent concepts.

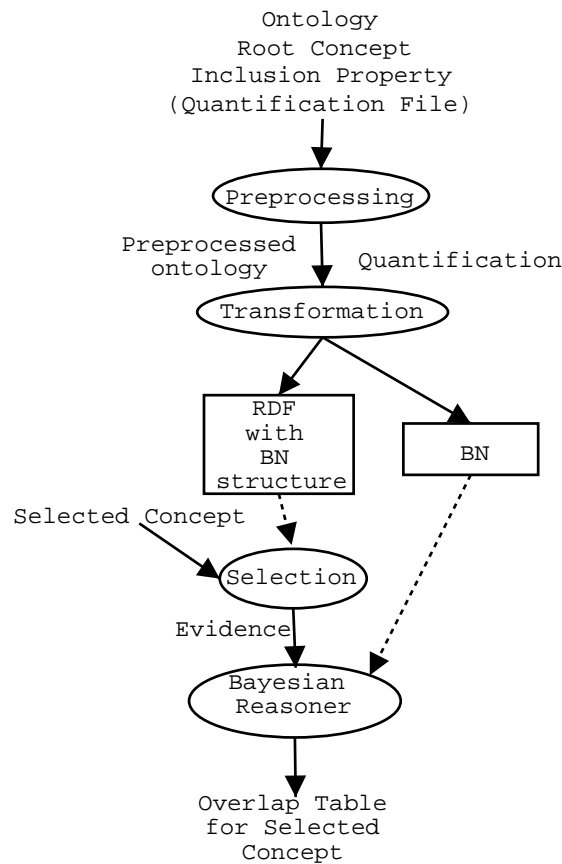


Figure 10.6: The architecture of the implementation.

The *preprocessing* module transforms the taxonomy into a predefined standard form. If an RDF data file that contains data records annotated according to the ontology is given as optional input, then the preprocessing module determines the mass of each concept in the taxonomy based on these annotations. The mass is the number of data records annotated to the concept directly or indirectly. The quantification principle is illustrated in Figure 10.7.

The *transformation* module implements the transformation algorithm, and defines the CPTs of the resulting Bayesian network. In addition to the Bayesian network, it creates an RDF graph with an identical topology, where nodes are classes and the arcs are represented by the *rdf:subClassOf* property. This graph will be used by the

selection module that expands the selection to include the concepts subsumed by the selected one, when using the Bayesian network.

The *Bayesian reasoner* does the evidence propagation based on the selection and the Bayesian network. The selection and Bayesian reasoner modules are operated in a loop, where each concept in the taxonomy is selected one after the other, and the overlap table is created.

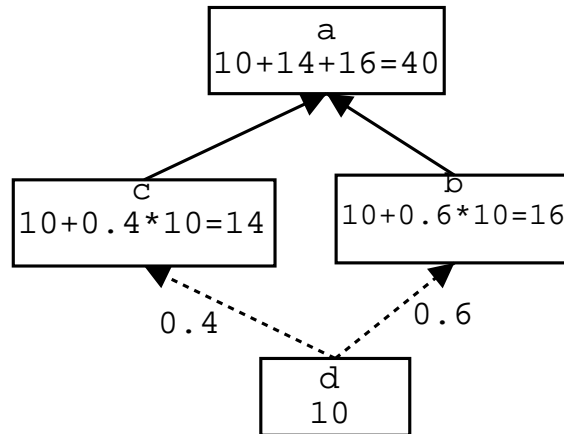


Figure 10.7: Quantification of concepts. The number of direct instances of each concept is 10. In the case of partial subsumption, only a part of the mass of the subconcept is taken as the mass of the superconcept

The *preprocessing*, *transformation*, and selection modules are implemented with SWI-Prolog²². The Semantic Web package is used. The *Bayesian reasoner* module is implemented in Java, and it uses the Hugin Lite 6.3²³ through its Java API.

10.6 Conclusions

Our method for modeling conceptual overlap between concepts proved to be simple, but it still enables the representation of overlap and partial subsumption between

²²<http://www.swi-prolog.org/>

²³<http://www.hugin.com/>

concepts. Our two approaches to compute the overlap values between concepts are useful, but in the implementation of the probabilistic faceted semantic search framework that will be presented in Chapter 11 we used the solid path structure approach and not the Bayesian network approach. This is because the computations were in practice more straight forward to implement using the taxonomy directly without the transformation to a Bayesian network.

Overlap graphs are simple and can be represented in RDF easily. Using the notation does not require knowledge of probability or set theory. The concepts can be quantified automatically, based on data records annotated according to the ontology, for example. The notation enables the representation of any Venn diagram, but there are set structures, which lead to complicated representations.

Such a situation arises, for example, when three or more concepts mutually partially overlap each other. In these situations some auxiliary concepts have to be used. We are considering to extend the notation so that this kind of situations could be represented better. On the other hand, taxonomies can be designed so that they avoid the extensive usage of partial overlap.

We used a geographical example case when presenting the method, however, the method is not limited to the geographical domain. In fact, in Chapter 11, when presenting the probabilistic faceted semantic search framework we will use examples from the mental health sector, and the approach fits also to other domains. As a minimalistic example, see Figure 10.8. This example represents the situation where the concept *Mirror* partially overlaps *Furniture*, *Car parts*, and *Personal belongings*, because some mirrors are furniture, some are parts of cars, and some are personal belongings that can be held in a person's pocket. Figure 10.9 presents the overlap graph of Figure 10.8 quantified based on the annotations and transformed into the solid path structure. Based on this solid path structure the overlap values between the concepts can be computed using Equation 10.1.

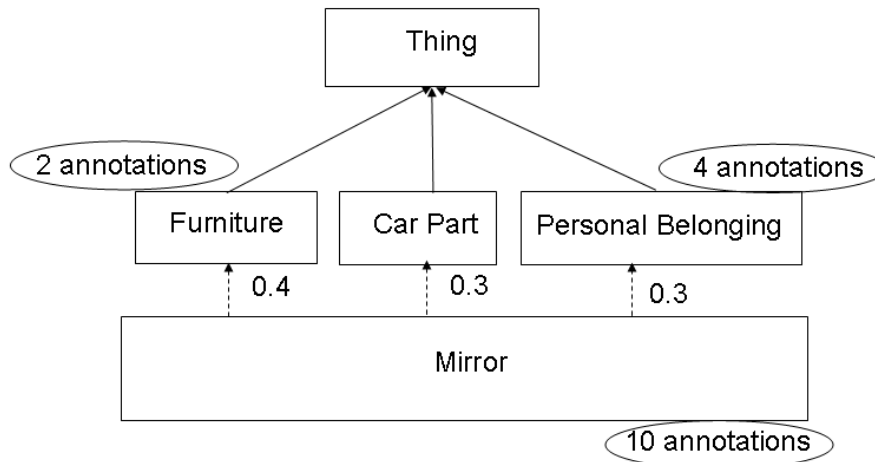


Figure 10.8: Overlap graph where mirror is partially overlapping furniture, car parts, and personal belongings

The Bayesian network structure that is created with the presented method is only one of the many possibilities. This one was chosen, because it can be used for computing the overlap tables in a most direct manner. The next section will discuss a number of other approaches to combine an ontology to a Bayesian network.

10.7 Related Work

Ng [85, 84] presents methods to combine probabilistic information with logic programming. This is called probabilistic logic programming. In principle we could have also created a probabilistic logic database for the taxonomy with Equation 10.1. However, this would be inefficient with large ontologies, because all the possible concept combinations would have to be taken into account and encoded in the database.

Ding and Peng [32] present principles and methods to convert an OWL ontology

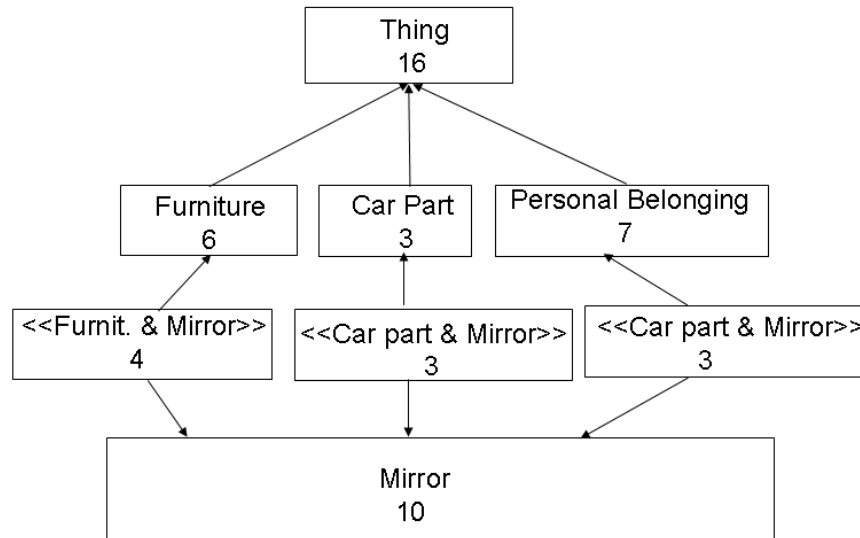


Figure 10.9: Overlap graph of Figure 10.8 quantified and transformed to solid path structure.

into a Bayesian network. Their methods are based on probabilistic extensions to description logics. For more information on these extensions, see [66, 39]. The approach has some differences to ours. First, their aim is to create a method to transform any OWL ontology into a Bayesian network. Our goal is not to transform existing ontologies into Bayesian networks, but to create a method by which overlap between concepts could be represented and computed from a taxonomic structure. However, we designed the overlap graph and its RDF implementation so that it is possible, quite easily, to convert an existing crisp taxonomy to our extended notation. Second, in the approach of Ding and Peng, probabilistic information must be added to the ontology by the human modeler that needs to know probability theory. In our approach, the taxonomies can be constructed without virtually any knowledge of probability theory or Bayesian networks.

Also other approaches for combining Bayesian networks and ontologies exist. Gu [41] present a Bayesian approach for dealing with uncertain contexts. In this approach probabilistic information is represented using OWL. Probabilities and conditional

probabilities are represented using classes constructed for these purposes. Mitra [79] presents a probabilistic ontology mapping tool. In this approach the nodes of the Bayesian network represents matches between pairs of classes in the two ontologies to be mapped. The arrows of the BN are dependencies between matches.

Kauppinen and Hyvönen [63] present a method for modeling partial overlap between versions of a concept that changes over long periods of time. The approach differs from ours in that we are interested in modeling degrees of overlap between different concepts in a single point of time.

11 Probabilistic Faceted Semantic Search

This chapter will present the *probabilistic faceted semantic search (PFSS)* framework, which will incorporate the method for modeling uncertainty presented in Chapter 10, weighted annotations, and separate end-user category definitions mapped to annotation ontologies to provide ranking of search results. *PFSS* also supports the combination of more than one ranking schemes to compute the probabilities of document relevance. Thus, *PFSS* aims to solve the problems encountered both in *CFSS*, and *FFSS*.

In *PFSS* the semantic knowledge base containing the document annotations, the annotation ontologies, and the search category definitions are interpreted probabilistically. The probabilities of document relevance in relation to each search category is computed using a recursive algorithm that adheres to the probabilistic interpretation of the knowledge base. The rest of this chapter is organized as follows:

1. *A motivating example.* Section 11.1 will present a motivating example of a *PFSS* search system and the knowledge base behind it. This motivating example will be used when the *PFSS* will be presented in the rest of the chapter.
2. *PFSS in a nutshell.* In Section 11.2 we will present the main intuitive principles behind *PFSS* as well as the high-level *PFSS* algorithm and the basic probability model. This section should give the reader a high-level understanding of *PFSS* to be able to follow the detailed presentation that is contained in the sections that follow.
3. *Simple search category.* Section 11.3 will present the case where probabilities of document relevance are computed for a search category that has a one-to-one mapping to an annotation concept. We will further develop

the probabilistic method for modeling overlap between concepts presented in Section 10 to include documents and search categories so that the desired probabilities of document relevance can be computed.

4. *Search categories defined in terms of Boolean combinations of annotation concepts.* In Section 11.4 the search categories are mapped to the annotation concepts using Boolean combinations (AND, OR, NOT). We will show how these kind of Boolean combinations can be incorporated into *PFSS*.
5. *Hierarchical search categories.* End-user search categories are typically organized as a hierarchy. In Section 11.5 we will describe how the hierarchical organization of the search categories is taken into account when computing probabilities of document relevance.
6. *Combining ranking schemes.* In Section 11.6 we will develop the framework further by allowing the usage of multiple ranking schemes simultaneously. The final probability of relevance of a document computed by *PFSS* is a mixture,—i.e., a weighted average—of the probability given by each ranking scheme.
7. *Performing the search.* After the search category specific probabilities of document relevance are computed they can be used to answer to a faceted search provided by the user as will be presented in Section 11.7.
8. *Summary.* Section 11.8 will summarize the results presented in this chapter.

For each detailed case presented in Sections 11.3–11.7 we will present the intuitive principles behind the case, and how the high-level algorithm and the probability model of Section 11.2 is extended in this case.

11.1 A Motivating Example

As a motivating example for *PFSS*, a hypothetical mental health portal is presented here. The portal is called *Virtual Mental Health Center (VMHC)*. It aims to provide mental health information to citizens as well as help non-expert users to find available mental health services based on their problem or need. To achieve the latter the portal offers a faceted search functionality in which the user can search for services based on typical mental health symptoms/problems, and the severity of the problem. A mockup user interface of *VMHC* is presented in Figure 11.1. The *Symptom* and *Severity* facets are shown on the left, the search results are presented as a list in the middle and they are also visualized on a map based on the addresses of the service provider. The search results are presented according to their relevance in a descending order.

The mental health service search is based on a semantic knowledge base in which semantic metadata of each service is stored. A sample from this knowledge base is presented in Figure 11.2. The services are annotated according to the *ICD-10* disease classification [2], a step vocabulary that indicates the level of severity of the mental health problems that the service deals with, and a small geo-spatial taxonomy. End-user facets are defined separately with end-users' needs in mind. To provide the search functionality depicted in Figure 11.1 these end-user facets are mapped onto the annotation concepts.

D2 and *D5* represent two of the services that are contained in the knowledge base. We will mostly refer to *D2* and *D5* as documents in the following text. The ovals represent annotation concepts, squares represent search categories, and the squares with soft corners represent Boolean combinations of concepts. The components of a Boolean concept are marked using the straight dashed arrows that originate from the Boolean concept. In the context of *PFSS* the semantics of each type of Boolean concept can be defined as follows:



Figure 11.1: A mockup of the Virtual Mental Health Center portal.

OR Documents that are relevant to at least one of the component concepts of this *OR* concepts are relevant to this concept. Notice, that the dashed arrows emanating from *OR* concepts, e.g. $Step1 \cup Step2$ can have weights in the range $[0, 1]$, and the default weight is 1.0. If the arrow is weighted, then probabilities of relevance according to the weighted component concept are trusted only with the probability indicated by the weight.

AND Documents that are relevant to all the component concepts of this *AND* concept are relevant to this concept.

NOT Documents that are not relevant to the component concept of this *NOT* concept are relevant to this concept.

The solid arrows between annotation concepts represent a *subconceptOf* relationship, i.e. transitive inclusion. The bended dashed arrows between documents and annotation concepts represent annotation relationship. The numbers next to these bended arrows represent annotation weight, i.e. how probable it is that the annotated document is relevant to the annotation concept. The bolded arrows between search categories and concepts represent the mappings of search categories to the annotation concepts. The numbers next to these bolded arrows represent the weight of the mapping. A mapping with weight 1.0 can be understood as an instruction to the *PFSS* system saying, "When determining probabilities of document relevance for this search category, trust with probability 1.0 to the probabilities of document relevance computed according to this—mapped—annotation concept". The weight is in the range $[0, 1]$, and the above described instruction is adapted accordingly. The arrows with crisp corners represent *subCategoryOf* relationships, i.e., the search category at the start of the arrow is a subcategory of the search category at the end of the arrow. This relationship is transitive.

Based on the above descriptions we can already make some intuitive estimations of probabilities of document relevance for search category selections. For example, consider the case that the user has selected the search category *Fears*. This search category is mapped to the annotation concept *F40 Phobias* with weight 1.0, which means that when *PFSS* determines the probabilities of document relevance for the search category *Fears* it completely trusts probabilities of document relevance computed for *F40 Phobias*. We see, that document *D2* is annotated to *F40 Phobias* with weight 0.9. Annotation weights in *PFSS* mean probability of document relevance. Thus, probability of document *D2* being relevant to *F40 Phobias* is 0.9. Thus, the probability of relevance of *D2* to the selected search category *Fears* is 0.9.

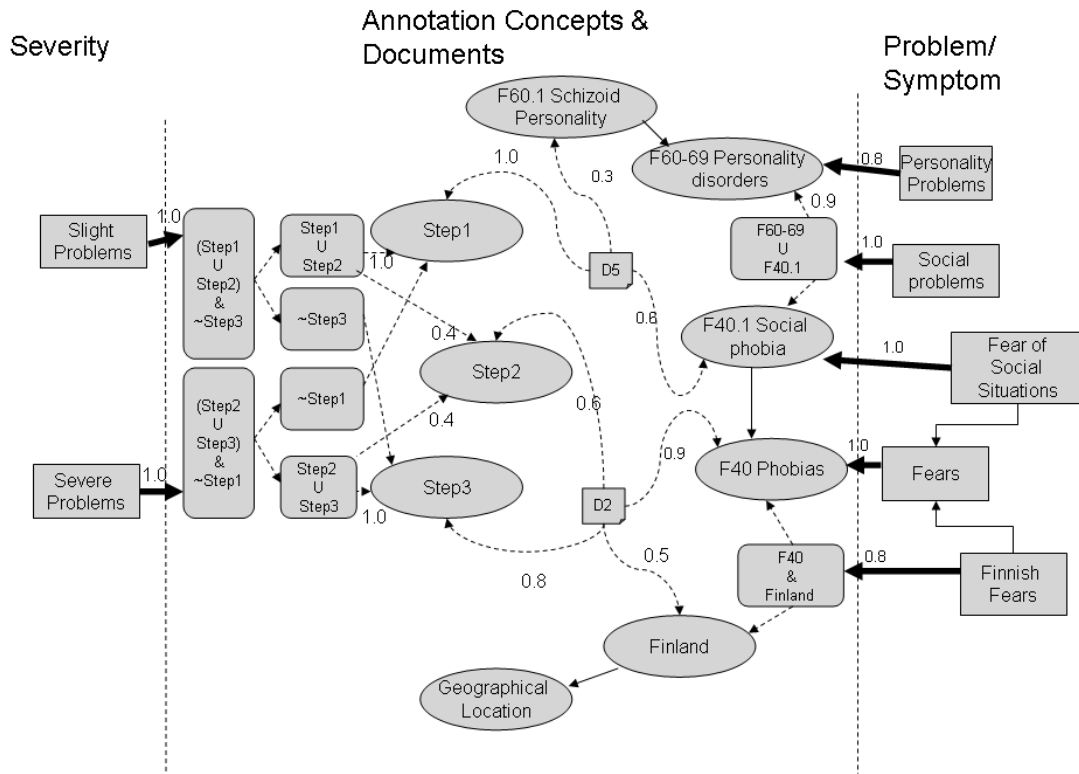


Figure 11.2: Conceptual structure behind the Virtual Mental Health Center search functionality.

11.2 PFSS in a Nutshell

This section presents the general intuition behind *PFSS*, the high-level algorithms for computing the probabilities of document relevance for facet selections, and the overview of the *PFSS* probability model. The details of the *PFSS* framework will be presented in Sections 11.3-11.7.

11.2.1 The Intuitive Interpretation of PFSS

In *CFSS* the result set of a multi-faceted search is the intersection of the result sets of the individual facets. We will follow this intuition in *PFSS*. In faceted search the facets are generally considered orthogonal, i.e. independent, of each other. Because the *product* is the probabilistic operator for computing the co-occurrence of two mutually independent events, in *PFSS* the probability of relevance of a document in relation to the facet selections of the user is the *product* of each individual facet selection specific probability.

The probability of relevance of a document in relation to a facet selection—i.e., a selection of a single search category—is computed recursively. This recursive computation is motivated by the structure and interpretation of the knowledge base described in Section 11.1: The weighted annotations are interpreted directly as probabilities of document relevance, and the relationships between the conceptual entities (annotation concepts, Boolean concepts, and search categories) are interpreted as instructions how to infer the probability of document relevance for one party in the relationship based on the probability of relevance of that document for the other party. This translates naturally into a recursive algorithm. The base case of the algorithm is the computation of the probabilities of document relevance in relation to the annotation concepts. The recursive step then uses the probabilities of document relevance specific to annotation concepts for computing the probabilities of document relevance specific to the selected search categories.

To clarify this, let us look at the example described in the last paragraph of the previous Section (11.1). In this example the base case is the computation of the probability of relevance of *D2* in relation to the annotation concept *F40 Phobias*, which yields the probability 0.9. This probability is then used to compute the probability of *D2* in relation to the selected search category *Fears*. This is the recursion step of the computation, which is performed based on the mapping of

the search category *Fears* to the annotation concept *F40 Phobias*. Because, in the example case, the mapping has the weight 1.0 the probability of relevance of *D2* to *Fears* is the same as the probability of relevance of *D2* to *F40 Phobias*.

The above example is very simple in the following respects: First, the probability of relevance of the document *D2* to the annotation concept *F40 Phobias* is acquired directly from the annotation weight in the knowledge base. Second, the selected search category is directly mapped to the annotation concept, which leaves us with a single recursion step. In *PFSS* both the base case and the recursion can be more complicated as follows:

Base case When computing the probability of relevance of documents to annotation concepts, not only direct but also indirect annotations are taken into account. For example, in Figure 11.2, the document *D5* is indirectly annotated to the concept *F60-69 Personality Disorders*, because the document is directly annotated to *F60.1 Schizoid Personality*, which is the subconcept of *F60-69 Personality Disorders*. Thus, the computation of probabilities of relevance of documents in relation to annotation concepts is not comprised merely of the reading of the annotation weights of each document, but it involves also computations based on the subconcept relationships between the annotation concepts.

Recursion step The recursion may have more layers than just one. For example, when computing the probability of relevance of document *D2* to the search category *Severe Problems* we would first compute the probability of relevance of *D2* to the annotation concepts *Step 1*, *Step 2*, and *Step 3*. This is the base case. Then we would use the probabilities computed in the base case to compute the probability of *D2* to the Boolean concepts $\neg\text{Step1}$, and $\text{Step2} \cup \text{Step3}$. This is the first layer of recursion. Each type of Boolean concept has its own mathematical function to perform the necessary computation.

The probabilities computed in the first recursion step are used to compute the probability of relevance of $D2$ to the Boolean concept $\neg Step1 \cup (Step2 \cup Step3)$. This is the second layer of recursion. And finally, the probability computed in the second recursion layer is used to compute the probability of $D2$ to the selected search category *Severe Problems*. If the search category had sub categories, we would also have to first compute the relevance of the document in relation to the child categories, and use that probability in the computation of the final probability value. The computation of probabilities of document relevance specific to the different types of Boolean concepts is described in Section 11.4, the handling of the hierarchical structure of facets is described in Section 11.5.

11.2.2 The High-level PFSS Algorithm

Based on the above, the main algorithm for computing the probabilities of document relevance in relation to user's facet selections is outlined in Algorithm 4.

<p>Data: Semantic knowledge base skb, Search Category \square $facetSelections$</p> <p>Result: Probability of document relevance \square $documentRelevances$</p> <pre> 1 documentRelevances = \emptyset ; 2 foreach Document $d \in skb$ do 3 Probability of document relevance $docRel = 1$; 4 foreach Search Category $sc \in facetSelections$ do 5 $docRel * =$ compute document relevance for the search category sc using 6 Algorithm 5 with input skb, sc, and d; 7 end 8 add $docRel$ to $documentRelevances$; 9 end </pre>

Algorithm 4: The main algorithm for computing probabilities of document relevances in *PFSS*

As input the algorithm gets the semantic knowledge base and the facet selections made by the user. As a result the algorithm returns the probability of document relevance for each document in the knowledge base. Between lines 2 and 8 the algorithm loops through all the documents in the semantic knowledge base. The probability of relevance *docRel* of the current document is first set to 1. Then the algorithm loops through the facet selections done by the user and multiplies the current value of *docRel* by the probability of relevance of that document to the current selected search category *sc*. The probability of relevance of the document to each search category is computed using Algorithm 5, which is explained below. After the probability of document relevance *docRel* is computed for the facet selections of the user, *docRel* is added to the set of returned probabilities of document relevance *documentRelevances*.

Algorithm 5 outlines the high-level recursive algorithm for computing the probability of relevance of a document in relation to a conceptual entity. In PFSS a conceptual entity is either an annotation concept, a Boolean combination of annotation concepts, or a search category. Algorithm 5 is a general "template algorithm" which applies to all types of conceptual entities found in the semantic knowledge base presented in Section 11.1.

As input the algorithm takes the semantic knowledge base *skb*, a conceptual entity *ce*, and a document *d*. The algorithm returns the probability of relevance *docRel* of document *d* in relation to the conceptual entity *ce*. The algorithm is composed of one *if-then-else* statement. The base case—i.e., the computation of probability of document relevance for an annotation concept—of the algorithm is presented in line 2 in the *then* clause of the *if-then-else* statement. The details of how the annotation concept specific probability of a document relevance is computed will be presented in Section 11.3.

The recursive step of the algorithm is presented in the *else* clause on lines 3 - 11.

1 2 3 4 5 6 7 8 9 10 11 12	<p>Data: Semantic knowledge base skb, Conceptual Entity ce, Document d</p> <p>Result: Probability of document relevance $docRel$ specific to the conceptual entity ce</p> <p>if ce is an annotation concept then</p> <p style="padding-left: 20px;">$docRel =$ compute the probability of d for the annotation concept ce</p> <p>else</p> <p style="padding-left: 20px;">Conceptual Entity \square $compCes =$ the conceptual entities that according to skb are to be used to compute the probability of document relevance specific to ce ;</p> <p style="padding-left: 20px;">Probability of document relevance \square $compRels = \emptyset$;</p> <p style="padding-left: 20px;">foreach Conceptual Entity $compCe \in compCes$ do</p> <p style="padding-left: 40px;">Probability of document relevance $compRel =$ call this algorithm (5) with input skb, $compCe$, and d;</p> <p style="padding-left: 40px;">add $compRel$ to $compRels$;</p> <p style="padding-left: 20px;">end</p> <p style="padding-left: 20px;">$docRel =$ compute the relevance of d specific to ce based on $compRels$ according to type of ce and the type of each $compCe \in compCes$;</p> <p>end</p> <p>$docRel =$ compute the weighted average of $docRel$ and relevance of d according to other possibly provided ranking schemes.</p>
-------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Algorithm 5: The general recursive algorithm for computing relevance of document in relation to a conceptual entity (an annotation concept, Boolean concept, or a search category).

On line 5, the set of conceptual entities $compCes$ (*component conceptual entities*) that according to the knowledge base skb should be used to infer document's d probability of relevance $docRel$ for ce are extracted from skb . This set is defined by the type of the conceptual entity ce . The types were presented in Section 11.1, and will be described in detail in Sections 11.3-11.4. For each conceptual entity $compCe$ in $compCes$ the algorithm recursively calls itself to compute the probability of relevance of document d to $compCe$. After the probability of relevance of d to each conceptual entity in $compCes$ is computed and saved in the set $compRels$ the algorithm uses these probabilities to compute the probability of relevance $docRel$ of

d to ce on line 10. This computation is also dependent on the type of ce and each $compCe$ of the used conceptual entities in $compCe$, and will be described in detail in the following sections.

If other ranking schemes in addition to $PFSS$ are in use and they provide rankings for the conceptual entity ce , line 12 computes the final probability of document relevance $docRel$ specific to ce by averaging over these competing ranking schemes. The averaging can essentially happen for any conceptual entity, however, it will be discussed and evaluated only on the search category level. Combining of ranking schemes will be discussed in Section 11.6 and evaluated in Chapter 13.

Similarly as in $CFSS$, where the documents can be projected onto the search categories in a batch process when the search system is constructed, also in $PFSS$ the probabilities of document relevance specific to search categories can be precomputed. Thus, the only online processing remains the operation where the different facet selections are combined to compute the final probabilities of document relevance, i.e. Algorithm 4. In this sense $PFSS$ does not add to the computational complexity of $CFSS$. Thus, the online computation cost of in terms of running time $O(n(D) * n(F))$, where $n(D)$, and $n(F)$ are the number of documents and facets respectively. The amount of facets is typically low, so the cost is in practice linear with respect to the amount of documents. In real world situation, the documents are efficiently indexed in a database, so the running time would be less than linear. As for memory consumption, The algorithm requires a *document – search category* matrix, the cost of which is $O(n(D) * n(SC))$.

11.2.3 The PFSS Probability Model

We create a probability model for the base case, such that each annotation concept is represented by a binary random variable AC_X , where AC_X^1 means that the user

is interested in the annotation concept X . In other words, documents relevant to X are also relevant for the user. AC_X^0 has the opposite meaning. Each document is represented by a binary random variable D_i , where D_i^1 means that the document Di is relevant to the user, and D_i^0 has the opposite meaning. The annotations of Di to the annotation concept Y with weight w is interpreted as the conditional probability specification

$$\begin{aligned} P(D_i^1|AC_Y^1) &= w \\ P(D_i^0|AC_Y^1) &= 1 - P(D_i^1|AC_Y^1) \end{aligned} \quad (11.1)$$

The handling of multiply annotated documents will be described in Section 11.3.

In our simple example case the annotation of $D2$ to $F40 Phobias$ with weight 0.9 in Figure 11.2 is interpreted in the base case as the conditional probability specification

$$\begin{aligned} P(D_2^1|AC_{F40 Phobias}^1) &= 0.9 \\ P(D_2^0|AC_{F40 Phobias}^1) &= 0.1 \end{aligned} \quad (11.2)$$

Probabilistically, the task of the base case, i.e., line 2 of Algorithm 5 is to compute the conditional probability $P(D_i^1|AC_X)$ for each document in the knowledge base, s.t. X is conceptual entity ce in question. To do that the conditional probabilities that are defined according to Equation 11.1 are used. If the document Di is annotated only to concept Y , then the conditional probability $P(D_i^1|AC_x)$ is defined as follows:

$$P(D_i^1|AC_X) = P(AC_Y|AC_X) * P(D_i^1|AC_Y^1) \quad (11.3)$$

Where $P(AC_Y|AC_X)$ indicates how relevant documents that match the concept Y are to a user interested in X . The computation of $P(AC_Y|AC_X)$ as well as the case of multiply annotated documents will be described in Section 11.3.

A search category Cat is represented by binary random variable SC_{Cat} , where SC_{Cat}^1

represents the event that the user is interested in Cat , and SC_{Cat}^0 represents the opposite event. Facets are interpreted as collections of search categories and are not represented by random variables in $PFSS$. Notice, that this is similar to $FFSS$ where facets are not explicitly included in the model.

A mapping between the search category Cat and the target concept X with weight w is interpreted as the probability

$$\begin{aligned} P(D_i^1|SC_{Cat}^1) &= w * P(D_i^1|AC_X^1) \\ P(D_i^0|SC_{Cat}^1) &= 1 - P(D_i^1|SC_{Cat}^1) \end{aligned} \quad (11.4)$$

for all documents in the semantic knowledge base. In our simple example, the search category is mapped directly to an annotation concept. It can be, however, mapped also to a Boolean combination of annotation concepts, which will be discussed in Section 11.4. Equation 11.4 applies to the situation that the search category Cat does not have any child categories. In Section 11.5 we will present the generalization of this equation to the situation that Cat does have child categories. $PFSS$ also contains a facility to combine evidence of relevance from external ranking schemes, The generalization of Equation 11.4 to the situation that multiple ranking schemes are used will be presented in Section 11.6.

For example, according to the mapping of the search category $Fear$ to the annotation concept $F40 Phobias$ with weight 1.0,

$$\begin{aligned} P(D_2^1|SC_{Fear}^1) &= 1.0 * P(D_2^1|AC_{F40 Phobias}^1) = 1.0 * 0.9 = 0.9 \\ P(D_2^0|SC_{Fear}^1) &= 0.1 \end{aligned} \quad (11.5)$$

The final probability of document relevance to a multifaceted search is then defined as the conditional probability:

$$\begin{aligned} P(D_i^1|Sel) &= \prod_{SC_{Cat} \in Sel} P(D_i^1|SC_{Cat}^1) \\ P(D_i^0|Sel) &= 1 - P(D_i^1|Sel) \end{aligned} \quad (11.6)$$

where Sel is the set random variables representing the selected search categories. Thus, the final probability of document relevance is the product of the search category specific relevances of the selected search categories.

11.3 Simple Search Category

Here we present a detailed treatment of the situation where the user has selected one search category that is mapped to a simple annotation concept (not a Boolean combination). Figure 11.3 presents this situation from GUI point-of-view. Currently we also "forget" the fact that the system contains other search categories that are organized to facets. So in effect in this case our system consists only of one search category that has a one-to-one mapping to a simple annotation concept. In the example of Figure 11.3, the user has selected the only existing search category *Fear of Social Situations* as the symptom of interest, and the matching documents $D5$ and $D2$ are shown in the result list in the center of the GUI. Figure 11.4 presents the part of the conceptual model of Figure 11.2 that is relevant to this example case.

As in the example case used in the previous Section (11.2) our recursive computation consists of the base case, and a simple recursion step. In the base case we have to compute the probability of relevance of the documents to the annotation concept onto which the search category is mapped. In the recursion step we use the annotation concept specific probabilities to compute the probability of relevance of the documents to the search category.

11.3.1 The Base Case for Documents with a Single Annotation

In this example the documents are $D5$ and $D2$, and the annotation concept onto which the search category *Fear of Social* is mapped is $F40.1$ *Social Phobia*. We

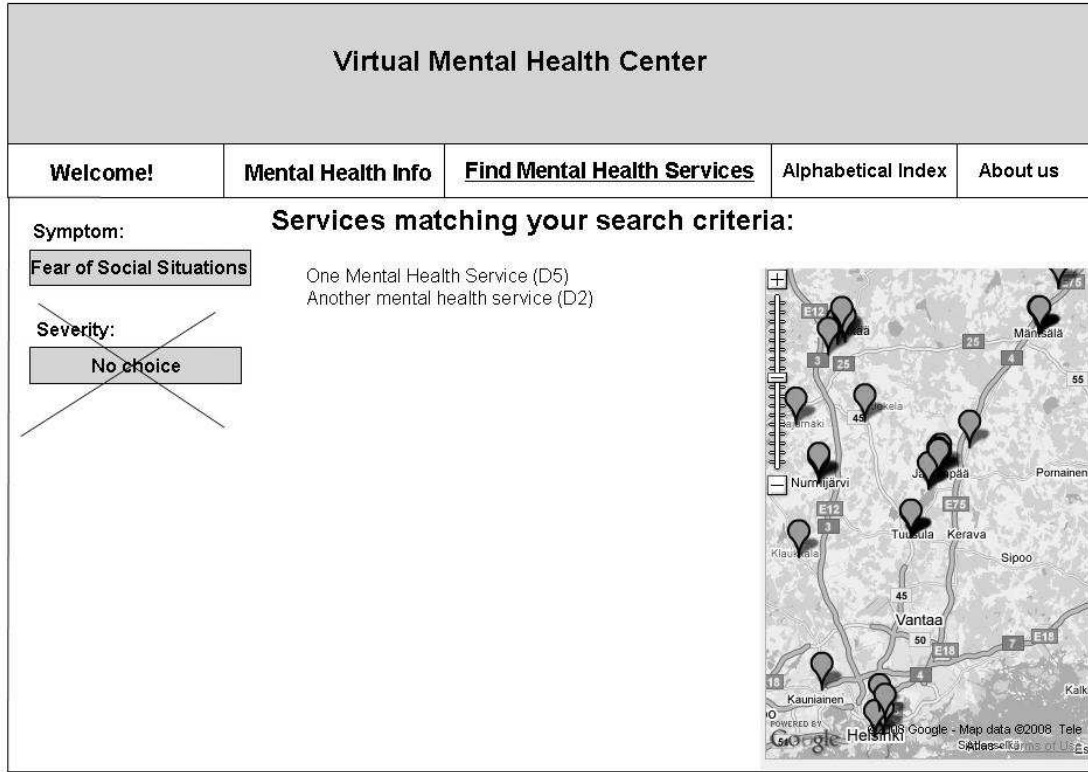


Figure 11.3: An example screenshot of a simple single faceted search.

see, that the document D_5 is annotated only to $F_{40.1}$ *Social Phobia* with weight 0.6, and D_2 is annotated only to F_{40} *Phobias* with weight 0.9. Thus, according to Equation 11.1 we can define the conditional probabilities

$$P(D_5^1 | AC_{F_{40.1} \text{ Social Phobia}}^1) = 0.6$$

$$P(D_5^0 | AC_{F_{40.1} \text{ Social Phobia}}^1) = 0.4$$
(11.7)

and

$$P(D_2^1 | AC_{F_{40} \text{ Phobias}}^1) = 0.9$$

$$P(D_2^0 | AC_{F_{40} \text{ Phobias}}^1) = 0.1$$
(11.8)

Because, the search category *Fear of Social Situations* is mapped to $F_{40.1}$ *Social Phobia* the task of the base case is to compute the probability of relevance of documents to $F_{40.1}$ *Social Phobia*, i.e., $P(D_5^1 | AC_{F_{40.1} \text{ Social Phobia}}^1)$, and

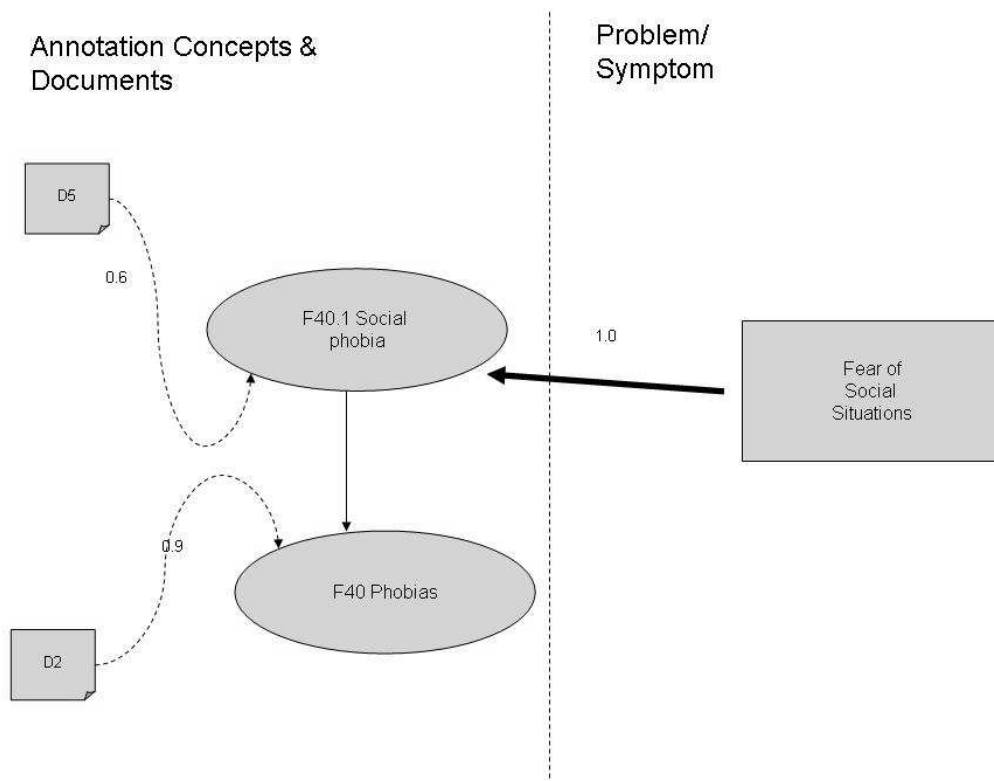


Figure 11.4: The conceptual model of the simple scenario for probabilistic faceted search.

$P(D_2^1 | AC_{F40.1 \text{ Social Phobia}})$. According to Equation 11.3 the conditional probability $P(D_5^1 | AC_{F40.1 \text{ Social Phobia}})$ is

$$\begin{aligned}
 P(D_5^1 | AC_{F40.1 \text{ Social Phobia}}) &= \\
 P(AC_{F40.1 \text{ Social Phobia}} | AC_{F40.1 \text{ Social Phobia}}) * & \\
 P(P(D_5^1 | AC_{F40.1 \text{ Social Phobia}})) &= \\
 1.0 * 0.6 &= 0.6
 \end{aligned}
 \tag{11.9}$$

To compute the marginal probability for D_2 we have to be able to specify the conditional probability $P(AC_{F40 \text{ Phobias}} | AC_{F40.1 \text{ Social Phobia}})$. Intuitively, we have to be able to specify the probability that a document matching $F40 \text{ Phobias}$ is relevant to a user who is known to be interested in $F40.1 \text{ Social Phobia}$. Recall, that the

method for computing overlap between concepts from Chapter 10 computes exactly probabilities of this kind. Thus, the *PFSS* algorithm uses the method of Chapter 10 to compute the probabilities of relevance of documents that are not annotated directly to the annotation concept that a search category is mapped to.

According to the method of Chapter 10, the concepts have to be given masses. In this case we compute the concept masses based on the annotations. Recall that quantification of concepts according to annotations was one of the options to create concept masses presented in Chapter 10. According to the annotation data the mass of *F40.1 Social Phobia* is 0.6, and the mass of *F40 Phobias* is $0.6 + 0.9 = 1.5$. *PFSS* does not expect the concept taxonomies to be represented using the *Overlap Graph* notation, but instead applies the solid path semantics to the crisp taxonomies. According to this, two concepts *A* and *B* intersect either if one is the subconcept of the other, or if *A* and *B* share one or more subconcepts. In the latter case the amount of overlap is based on the aggregated mass of the shared subconcepts. Otherwise, the concepts are interpreted to be disjoint.

After the concepts are massified the needed conditional probability $P(AC_{F40 Phobias}^1 | AC_{F40.1 Social Phobia}^1)$ can be computed using the formula:

$$\begin{aligned} P(AC_{F40 Phobias}^1 | AC_{F40.1 Social Phobia}^1) &= \\ m(F40 Phobias \cap F40.1 Social Phobias) / m(F40 Phobias) &= \quad (11.10) \\ 0.6 / 1.5 &= 0.4 \end{aligned}$$

Now we can compute $P(D_2^1 | AC_{F40.1 Social Phobia}^1)$ using Equation 11.3:

$$\begin{aligned} P(D_2^1 | AC_{F40.1 Social Phobia}^1) &= \\ P(AC_{F40 Phobias}^1 | AC_{F40.1 Social Phobias}^1) * P(D_2^1 | AC_{F40 Phobias}^1) &= \quad (11.11) \\ 0.4 * 0.9 &= 0.36 \end{aligned}$$

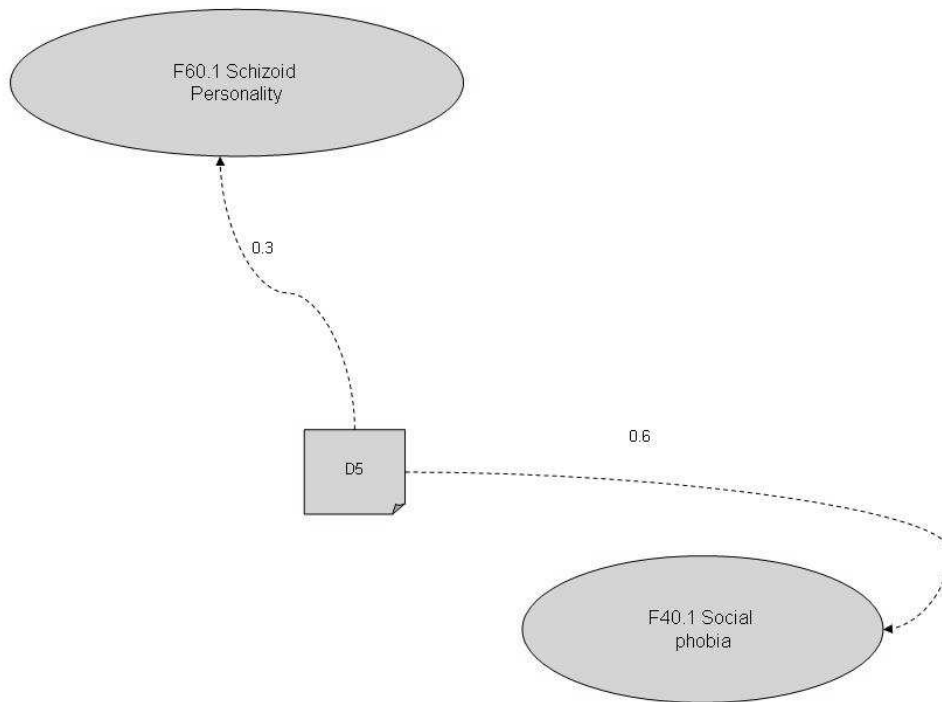


Figure 11.5: Multiply annotated document.

11.3.2 The Base Case for Documents with Multiple Annotations

Often documents are multiply annotated, as is exemplified in Figure 11.5. In the case of a multiply annotated document, we use the *noisy OR-Gate* [83] to combine the evidence from the individual annotations.

The noisy OR-gate is typically used when modeling relationships between variables each of which has only two values. In our case the variables are the document variable D_i , and the variables representing the concepts that D_i is annotated to, i.e., $AC_{par_1}, \dots, AC_{par_k}$. We will call these variables the parents of D_i . In noisy OR-Gate terminology D_i could be called the effect variable, and the parent variables

Table 11.1: The conditional probability table for the document variable D_5 , according to the annotation of Figure 11.5.

$P(D_5^1 AC_{F60.1}^1 \text{ Schizoid Personality}, AC_{F40.1}^1 \text{ Social Phobia}) = 1 - (1 - 0.3)(1 - 0.6) = 0.72,$
$P(D_5^0 AC_{F60.1}^1 \text{ Schizoid Personality}, AC_{F40.1}^1 \text{ Social Phobia}) = 1 - 0.72 = 0.28$
$P(D_5^1 AC_{F60.1}^1 \text{ Schizoid Personality}, AC_{F40.1}^0 \text{ Social Phobia}) = 1 - (1 - 0.3) = 0.3,$
$P(D_5^0 AC_{F60.1}^1 \text{ Schizoid Personality}, AC_{F40.1}^0 \text{ Social Phobia}) = 1 - 0.3 = 0.7$
$P(D_5^1 AC_{F60.1}^0 \text{ Schizoid Personality}, AC_{F40.1}^1 \text{ Social Phobia}) = 1 - (1 - 0.6) = 0.6,$
$P(D_5^0 AC_{F60.1}^0 \text{ Schizoid Personality}, AC_{F40.1}^1 \text{ Social Phobia}) = 1 - 0.6 = 0.4$
$P(D_5^1 AC_{F60.1}^0 \text{ Schizoid Personality}, AC_{F40.1}^0 \text{ Social Phobia}) = 0,$
$P(D_5^0 AC_{F60.1}^0 \text{ Schizoid Personality}, AC_{F40.1}^0 \text{ Social Phobia}) = 1 - 0 = 1$

could be called the cause variables. When using the noisy OR-Gate, the following three assumptions are made in the model [83]:

Inhibition There is some mechanism that inhibits a parent event from bringing about its effect, and the presence of the parent event bring about the effect only if this inhibition mechanism is disabled. In our case the parent event is the interest of the user in an annotation concept, which we know based on the search that the user specified, and the effect is the relevance of D_i to the user. The inhibiting mechanism is the possible situation that for some reason D_i does not contain useful information for the user that is interested in the annotation concept even though the document is annotated to that concept. This is why the gate is called *noisy*.

Exception independence This assumption entails that the mechanism that inhibits one parent event is independent of the mechanism that inhibits other parents. In our case, the case that D_i is not relevant to AC_{Par_1} does not mean that D_i is also not relevant to AC_{Par_2} .

Accountability This assumptions entails that an effect can happen— D_i is rel-

evant for the user—if at least one of its parent events is present and is not being inhibited. This is why the gate is called an *OR-Gate*, and it complies well with the functioning of *CFSS* where a document is considered relevant if at least one of its annotations matches the query. In the case of *PFSS* the annotation weight is interpreted as the probability that the inhibition is *not* present for the annotated concept.

Let $w_{Di \rightarrow J}$ represent the annotation weight of Di to annotation concept J , let Par_{D_i} denote the parents of D_i , and let $Par_{D_i}^1$ denote those parents AC_J that are in state AC_J^1 . Based on the above assumptions, the noisy OR-Gate for document variables is then computed as:

$$P(D_i^1 | Par_{D_i}) = 1 - \prod_{\forall AC_J \in Par_{D_i}^1} (1 - w_{Di \rightarrow J}) \quad (11.12)$$

for all documents Di in the knowledge base. Table 11.1 presents the conditional probabilities for the conceptual model of Figure 11.5 given the different combinations of assignments to the annotation concept variables. The conditional probability $P(D_i^1 | AC_X)$ for the base case is then:

$$P(D_i^1 | AC_X) = 1 - \prod_{\forall AC_J \in Par_{D_i}} (1 - P(AC_J | AC_X) * w_{Di \rightarrow J}) \quad (11.13)$$

for all documents Di in the knowledge base.

Notice, that Equation 11.3 is a special case of Equation 11.13, in the situation that Di is annotated only to one concept. Thus, Equation 11.13 can be used instead of 11.3 also in the case of a document with a single annotation. This concludes the presentation of the probability model for the base case of our algorithm.

11.3.3 The Recursion Step

The recursion step uses the probabilities of document relevance computed in the base case to compute the search category specific probabilities. In the case of simple search category definitions, the probability of relevance of documents can be computed simply using Equation 11.4. Notice, that the weight w of the mapping of *Fear of Social Situations* to *F40.1 Social Phobia* is 1.0. Thus, the probability of relevance of D_5 for the search category *Fear of Social Situations* is

$$\begin{aligned} P(D_5^1 | SC_{Fear\ of\ Social\ Situations}^1) &= \\ w * P(D_5^1 | AC_{F40.1\ Social\ Phobia}) &= \\ 1.0 * 0.6 &= 0.6 \end{aligned} \tag{11.14}$$

and the probability of relevance of D_2 is

$$\begin{aligned} P(D_2^1 | SC_{Fear\ of\ Social\ Situations}^1) &= \\ w * P(D_2^1 | AC_{F40.1\ Social\ Phobia}) &= \\ 1.0 * 0.36 &= 0.36 \end{aligned} \tag{11.15}$$

Because, *Fear of Social Situation* is the only selected search category, the search category specific probability of document relevance is also the final probability of relevance for the search according to Equation 11.6.

11.3.4 The Algorithm

This section presents the specialization of Algorithm 5 according to the case treated in this section; i.e., how the base case (line 2 of Algorithm 5) is computed, and then how this annotation concept specific probability of document relevance is used to compute the search category specific probability (line 10 of Algorithm 5):

1. **The Base Case (line 2 of Algorithm 5):**

- For each $ac \in SKB$ compute $P(ac|ce)$ according to the method described in Chapter 10.
- Compute the probability of relevance of d based on the conditional probabilities computed in the above step, and the annotation weights of d according to the noisy OR-Gate as in Equation 11.13.

2. **The Recursion Step (line 10 of Algorithm 5):**

- $docRel$ = annotation concept specific probability of relevance of d —as computed in the base case above—multiplied by the mapping weight between the selected search category and the annotation concept.

11.4 Search Categories Mapped to Boolean Combinations of Annotation Concepts

In this section we will show, how Boolean combinations of annotation concepts are handled in *PFSS*. The Boolean combinations of concepts that are supported by *PFSS* are *OR*, *AND*, and *NOT*.

11.4.1 OR

In Figure 11.6 the search category *Social Problems* is mapped to a Boolean combination *OR* of the annotation concepts *F40.1 Social Phobia* and *F60-69 Personality Disorders*. This is an example of a search category that is defined using a Boolean combination of annotation concepts. According to the semantics of Boolean combination concepts given in Section 11.1, documents that are relevant to either *F40.1 Social Phobia*, *F60-69 Personality Disorders*, or both are also relevant to the Boolean

combination $F40.1 \cup F60 - 69$. To compute the probability of relevance of document $D2$ to the selected search category *Social Problems* it should first compute the probability of relevance of the document to $F40.1$ *Social Phobia*, and to $F60-69$ *Personality Disorders*, after this determine the probability of relevance of $D2$ to $F40.1 \cup F60 - 69$, as a probabilistic *OR* combination of the component specific probabilities, and then, finally, compute the probability of relevance of $D2$ to *Social Problems* based on the $F40.1 \cup F60 - 69$ specific probability.

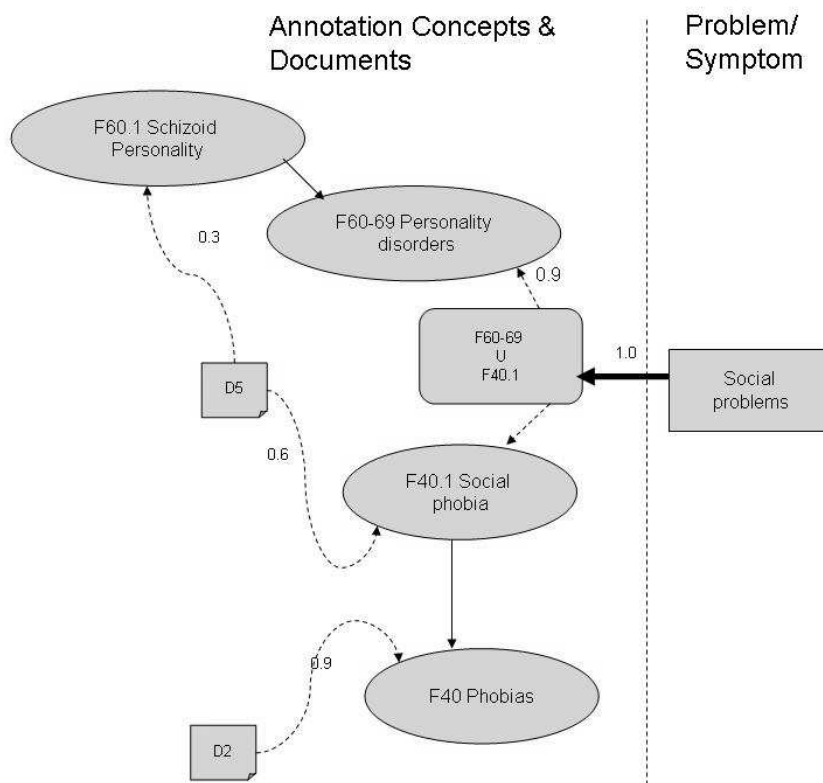


Figure 11.6: The search category *Social Problems* is defined using the Boolean combination *OR* of annotation concepts $F40.1$ *Social Phobia* and $F60-69$ *Personality Disorders*.

Notice, that the arrow from the Boolean combination $F60-69 \cup F40.1$ to the annotation concept $F60-69$ *Personality Disorders* has a weight 0.9. The intuitive meaning of this, in terms of probability of document relevance, is that we trust with proba-

bility 0.9 that documents relevant to *F60-69 Personality Disorders* are also relevant to $F60-69 \cup F40.1$. We can see, that the noisy OR-Gate used in combining evidence about multiple annotations is suitable to be applied to this situation to compute the *OR* combination specific probabilities of document relevance. D_i is still the effect variable, but now the inferred relevance of D_i to the component concepts are the parent events—instead of the interest of the user in an annotation concept as in Equation 11.13—and instead of an annotation weight we have the weight of the component concept in the OR combination. Thus, the *OR* combination is probabilistically interpreted as follows:

$$P(D_i^1 | OR_{Comp}^1) = 1 - \prod_{J \in Comp} (1 - P(D_i^1 | Comp_J^1) * w_{\cup Comp \rightarrow J}) \quad (11.16)$$

$$P(D_i^0 | OR_{Comp}^1) = 1 - P(D_i^1 | OR_{Comp}^1)$$

where

- $\cup Comp$ is the *OR* combination under inspection,
- OR_{Comp} is the binary random variable representing $\cup Comp$, s.t. OR_{Comp}^1 represents the situation that the user is interested in the *OR* combination, and OR_{Comp}^0 represents the opposite,
- $Comp$ is the set of component concepts that form $\cup Comp$,
- $Comp_J$ is a random variable representing the component concept J of $\cup Comp$. J might be an annotation concept or a Boolean combination of annotation concepts.
- $P(D_i^1 | OR_{Comp}^1)$ is the probability of relevance of document Di to a user that is interested in $\cup Comp$, and

- $w_{\cup Comp \rightarrow J}$ is the weight of the concept J in $\cup Comp$.

In the example of 11.6 the *OR* concept has only annotation concepts as its components. However, as will be shown in Section 11.4.3, the Boolean combination concepts can also have other Boolean combinations as components.

For example, Using Equation 11.16 we can compute the probability of relevance of documents $D2$ to the *OR* concept $F60-69 \cup F40.1$ according to Figure 11.6:

$$\begin{aligned}
P(D_2^1 | OR_{F60-69 \cup F40.1}^1) &= \\
1 - (1 - P(D_2^1 | AC_{F40.1}^1 \textit{ Social Phobia}) * w_{F60-69 \cup F40.1 \rightarrow F40.1}) * & \quad (11.17) \\
(1 - P(D_2^1 | AC_{F60-69}^1 \textit{ Personality Disorders}) * w_{F60-69 \cup F40.1 \rightarrow F60-69}) &= \\
1 - (1 - 0.36 * 1) * (1 - 0 * 0.9) &= 1 - 0.64 = 0.36
\end{aligned}$$

Notice, that the value $P(D_2^1 | AC_{F40.1}^1 \textit{ Social Phobia}) = 0.36$ was computed in Equation 11.11, and $D2$ is not annotated to *F60-69 Personality Disorders*, so $P(D_2^1 | AC_{F60-69}^1 \textit{ Personality Disorders}) = 0$. The probability of relevance for $D5$ can be computed similarly, but this computation is not explicitly shown here.

After $P(D_i^1 | OR_{Comp}^1)$ is computed, it is used to compute the search category specific probability as shown in Equation 11.4, such that the concept X of that equation is the *OR* combination concept which the search category is mapped to. For example the probability of $D2$ to the search category *Social Problems* is:

$$P(D_2^1 | SC_{\textit{ Social Problems}}^1) = 1.0 * P(D_2^1 | OR_{F60-69 \cup F40.1}^1) = 1.0 * 0.36 = 0.36 \quad (11.18)$$

11.4.2 AND

In Figure 11.7 the search category *Finnish Fears* is mapped to the *AND* combination of the annotation concepts *Finland* and *F40 Phobias*. According to the semantics

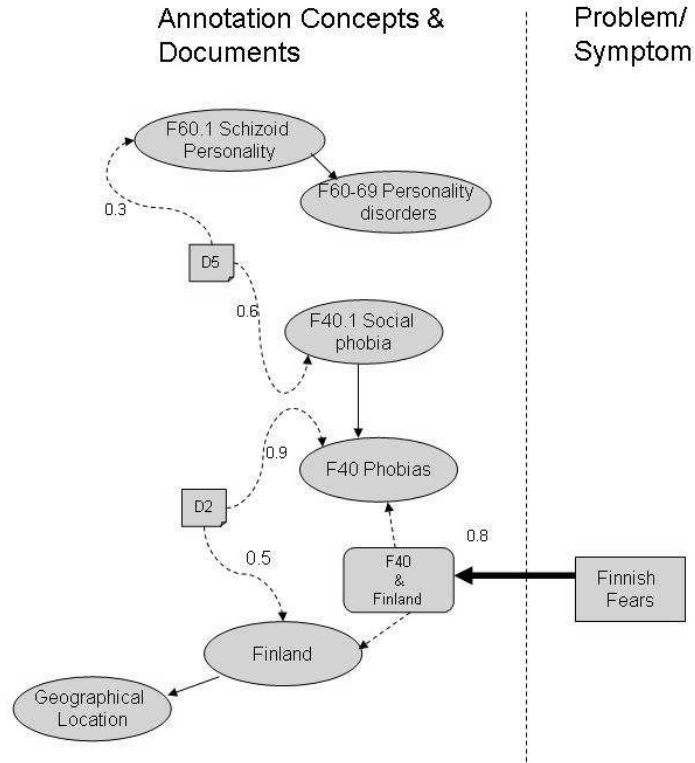


Figure 11.7: The search category *Finnish Fears* is defined using the Boolean combination *AND* of annotation concepts *Finland* and *F40 Phobias*.

given in Section 11.1, a document is relevant to the combination $F40 \cap Finland$, if it is relevant both to *Finland* and *F40 Phobias*. This correspond probabilistically to the product operation, which is used to compute the co-occurrence of two or more mutually independent events. Thus, the *AND* combination is interpreted as follows:

$$P(D_i^1 | AND_{Comp}^1) = \prod_{J \in Comp} P(D_i^1 | Comp_J^1) \quad (11.19)$$

$$P(D_i^0 | AND_{Comp}^1) = 1 - P(D_i^1 | AND_{Comp}^1)$$

where $P(D_i^1 | AND_{Comp}^1)$ is the probability of relevance of document D_i to a user that is interested in the *AND* combination concept $\cap Comp$, and $Comp$ is the set of component concepts that form $\cap Comp$. As in the *OR* combination, each combination concept might be an annotation concept or a Boolean combination.

The probabilities of relevance for $D2$ and $D5$ to $F40 \cap Finland$ are thus:

$$\begin{aligned} P(D_2^1 | AND_{F40 \cap Finland}^1) &= \\ P(D_2^1 | AC_{F40 Phobias}^1) * P(D_2^1 | AC_{Finland}^1) &= \quad (11.20) \\ 0.9 * 0.5 &= 0.45 \end{aligned}$$

and

$$\begin{aligned} P(D_5^1 | AND_{F40 \cap Finland}^1) &= \\ P(D_5^1 | AC_{F40 Phobias}^1) * P(D_5^1 | AC_{Finland}^1) &= \quad (11.21) \\ 0.6 * 0 &= 0 \end{aligned}$$

The computation of the annotation concept specific probabilities is done according to Equation 11.13, but is not explicitly shown here.

As discussed in Section 11.4.1 the search category specific probabilities are computed using Equation 11.4. For example, the probability of relevance of $D2$ to the search category *Finnish Fears* is:

$$P(D_2^1 | SC_{Finnish Fears}^1) = 0.8 * P(D_2^1 | AND_{F40 \cap Finland}^1) = 0.8 * 0.45 = 0.36 \quad (11.22)$$

Notice the effect of the mapping weight, which is 0.8. For $D5$ the search category specific probability is naturally 0 because $P(D_5^1 | AND_{F40 \cap Finland}^1) = 0$.

11.4.3 NOT

The *NOT* operator is used in the definition of the search category *Slight Problems*. According to semantics of Section 11.1 a document is relevant to $(Step1 \cup Step2) \cap \neg Step3$ if it is relevant to *Step 1*, or *Step 2*, but not *Step 3*. This definition of *Slight Problems* shows that the Boolean combinations of concepts can themselves be components of other Boolean combinations. Search category definitions can have a recursive structure, i.e., they can be defined using Boolean combinations of Boolean combinations and so forth.

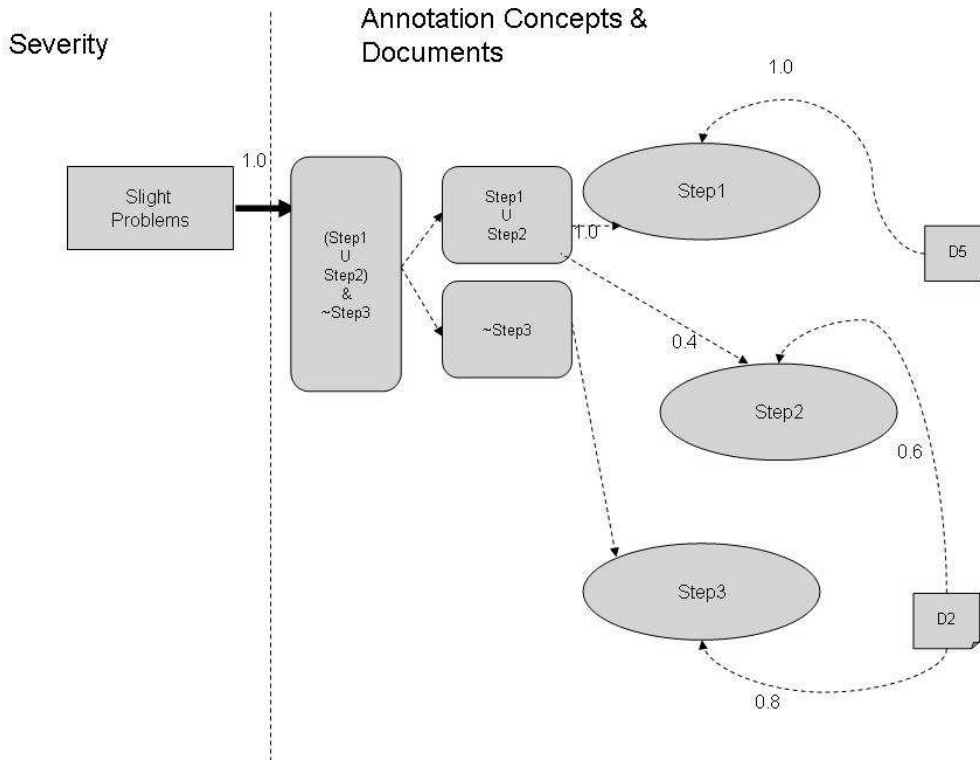


Figure 11.8: The search category *Slight Problems* uses *OR*, *AND*, and *NOT* Boolean combinations of annotation concepts in its definition.

Probabilistically the *NOT* concept is interpreted as:

$$P(D_i^1 | NOT_j^1) = P(D_i^0 | AC_j^1) \quad (11.23)$$

i.e., the probability that D_i is relevant to the negation of concept J equals the probability that D_i is not relevant to the concept J . Again, J might be an annotation concept or a Boolean combination concept.

For example, the probability of relevance of D_2 to $\neg Step3$ is:

$$P(D_2^1 | Not_{Step3}^1) = P(D_2^0 | AC_{Step3}^1) = 1 - 0.8 = 0.2 \quad (11.24)$$

To compute the relevance of D_2 and D_5 to the search category *Slight Problems* we

would do the following:

1. Compute the relevance of $D2$ and $D5$ to all three annotation concepts in Figure 11.8.
2. Use the annotation concept specific probabilities to compute the $\neg Step3$ and $Step1 \cup Step2$ specific probabilities.
3. Use the $\neg Step3$ and $Step1 \cup Step2$ specific probabilities to compute the $(Step1 \cup Step2) \cap \neg Step2$ specific probability.
4. Finally, use the $(Step1 \cup Step2) \cap \neg Step2$ specific probability to compute the *Slight Problems* specific probability.

Notice, that the Boolean combinations *OR*, *AND*, and *NOT* as interpreted in *PFSS* are different from *Union*, *Intersection*, and *Complement* concepts, respectively, as defined e.g. in *OWL*. In *PFSS* one can define a search category as the *AND* of two disjoint concepts, e.g. *Finland* and *Russia*, in the example of Chapter 10, and this will yield valid search results, but the *Intersection* of *Finland*, and *Russia* does not exist. This is because the search category definitions refer to sets of relevant documents, which can be discovered using the annotation concepts, but *OWL* refers to the concepts as such.

11.4.4 The Algorithm

Based on the above, line 10 (the recursive step) of Algorithm 5 is extended in the case of a Boolean combination as follows:

1. If the conceptual entity ce is an *OR* combinations:
 - $docRel$ = the noisy OR computed from the probabilities of document relevance $compRel$ s of the component conceptual entities $compCe$ of ce .
2. If the conceptual entity ce is an *AND* combination:
 - $docRel$ = the product of the probabilities of document relevance $compRel$ s to the component conceptual entities $compCe$ of ce .
3. If the conceptual entity ce is a *NOT* combination:
 - $docRel = 1 - compRel$, i.e. probability of relevance of d is the probability of relevance $compRel$ of d specific to the component conceptual entity $compCe$ subtracted from 1.

11.5 Hierarchical Search Categories

Figure 11.9 presents a hierarchy of search categories. The search category *Fears* is the parent of the search categories *Fear of Social Situations*, and *Finnish Fears*. According to crisp faceted semantic search, if a document matches a search category then it also matches all the ancestor search categories. The opposite, however, is not true, i.e., if a document matches a search category it does not by default match the descendant search categories. This intuition is followed in *PFSS* in the following way:

1. The fact that a document matches a descendant of the selected search category is seen as evidence that the document is relevant. However, in *PFSS* a direct match to the selected search category is considered better evidence. The longer the path between the selected and the matching search category the less certain *PFSS* is about the relevance.

2. Also in *PFSS* a match to an ancestor of the selected search category is not seen as evidence that the document is relevant to the user.

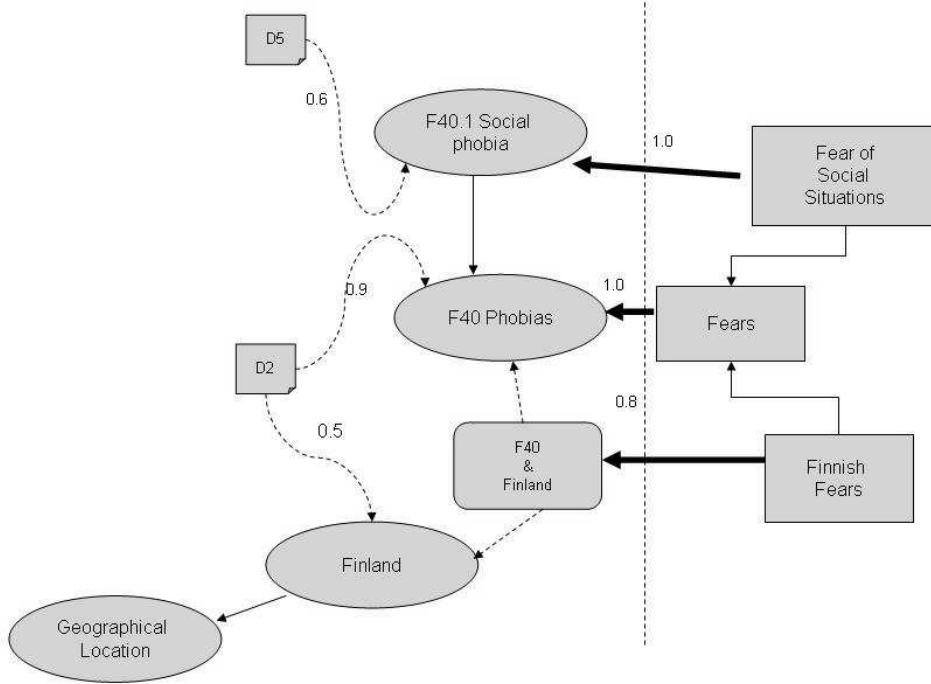


Figure 11.9: Hierarchical organization of search categories.

Thus, if a search category has subcategories, the probability of document relevance is computed based on the mapped concept (or combination of concepts) as defined in Equation 11.4 and the child search categories using the noisy OR-Gate:

$$P(D_i^1 | SC_{Cat}^1) = 1 - \prod_{\forall Child_j \in Child_{Cat}} (1 - sn * P(D_i^1 | SC_{Child_j}^1)) * (1 - w * P(D_i^1 | X)) \quad (11.25)$$

where

1. $Child_{Cat}$ is the set of child categories of Cat ,

2. $Child_j$ represents one child search category of Cat , and $SC_{Child_j}^1$ is the random variable representing its interestingness to the user,
3. X is the concept that Cat is mapped to with weight w , and
4. sn is the probability that a document is relevant to the selected search category, if the document is relevant to its child category. In our implementation we have set it manually to 0.8. This reflects the intuition that a direct match to a search category is better evidence than a match to a subcategory, and the longer the hierarchical path the less certainty about the evidence,

As can be seen, Equation 11.4 is a special case of Equation 11.25 in the situation that Sc does not have any child categories.

Based on the above, the recursion step (line 10 of Algorithm 5) is handled in the case of a search category that might contain child categories as follows:

- $docRel$ = noisy OR of the probability of relevance of document d specific to the conceptual entity that the search category ce is mapped to and each of the sub categories of ce such that firstly, the trust in the mapped conceptual entity is set according to the mapping weight and secondly, the trust in each of the child categories is 0.8.

11.6 Combining Evidence of Multiple Ranking Schemes

We will now extend $PFSS$ by combining multiple ranking schemes to the model. We want to be able to combine different ranking schemes, because it has been shown that combining evidence about relevance improves ranking [70]. In addition, we acknowledge the fact that there are many—and sometimes contradicting—ways

to utilize the conceptual knowledge in the semantic knowledge base for ranking of search results. Any ranking schemes can be combined to *PFSS* as long as the relevance values are normalized to the range $[0, 1]$ so that they can be interpreted as probabilities. In chapter 13 we will evaluate the effect of combining evidence by combining the rankings provided by *CFSS*, *FFSS*, and a heuristic ranking method to *PFSS* using the method described in this section.

Our aim is to combine the probabilities of relevance computed by the different ranking schemes in a probabilistically sound way. The probabilistic way to combine competing distributions is by creating a mixture distribution, i.e., a weighted average of the competing distributions as follows:

$$P(D_i^1 | SC_{Cat}^1) = \sum_{\forall rs_j \in RS} w_{rs_j} * P(D_i^1 | SC_{Cat:rs_j}^1) \quad (11.26)$$

where

1. RS is the set of ranking schemes that are used,
2. rs_j is one of those ranking schemes,
3. w_{rs_j} is the weight of the ranking scheme rs_j . The sum of the weights of all ranking schemes equals 1,
4. $P(D_i^1 | SC_{Cat:rs_j}^1)$ is the probability of relevance of document Di to the search category Cat according to the ranking scheme rs_j .

The weight of each scheme can be configured in the *PFSS* system, but if we lack better knowledge we can set each value the probability of $1/n(RS)$, where $n(RS)$ is the number of ranking schemes, according to the principle of indifference. This probability could also be learned, however, currently *PFSS* does not contain a facility for learning probabilities.

Thus, the averaging of ranking schemes (line 12 of Algorithm 5) is done as follows:

- $docRel$ = the weighted average of $docRel$ according to $pfss$ and other possibly provided ranking schemes computed according to Equation 11.26.

11.7 Performing the Search

We have now presented the complete algorithm and probability model for computing the probabilities of document relevance in relation to a search facet. After these search category specific probabilities are computed, they are combined to answer to a multifaceted search given by the user.

Recall from Section 11.2, that in *PFSS* the probabilities of relevance of documents in relation to the facet selections of the user are computed as the product of the probabilities in relation to each selected search category. This is, in fact, similar to the *AND* operator that was described on Section 11.4. Algorithm 4 presented the computation of probabilities of relevance of documents in relation to a multi-facet search given by the user, and Equation 11.6 presented the probabilistic interpretation of this computation.

11.8 Summary

This chapter presented the *probabilistic faceted semantic search (PFSS)* framework, including its intuitive principles, the probability model, and the algorithms for constructing the model and computing the documents of relevance in response for searches specified by the user. The recursive structure of the model enables efficient performance, because it enables us to precompute most of the model. In effect, the

only online operation is the combination of facet specific relevances in the multi-faceted search scenario.

The next Chapter (12) will summarize the lessons learned of the development of the framework, and the contributions of the probabilistic approach to the faceted semantic search developed in this part of the dissertation. Chapter 13 will then present an empirical comparison and evaluation of *CFSS*, *FFSS*, and *PFSS*, and will draw conclusions based on the comparison. Appendix A presents the technical design of a reference implementation of the *PFSS* framework, which contains an RDF-based language for representing the search facets, the Boolean combinations of concepts, and the weighted annotations. The system is implemented in the Java²⁴ programming language. The implementation described in Appendix A was used in the evaluation that will be presented in Chapter 13.

²⁴<http://java.sun.com/>

12 Contributions and Lessons Learned

This chapter will summarize the contributions of the *Probabilistic Approach*. After the contributions are present lessons learned are discussed. The next section will then provide an empirical comparison of *CFSS*, *FFSS* and *PFSS*.

12.1 Contributions

The main contributions of the probabilistic approach presented in this part of the dissertation are the following:

1. **Modeling of Uncertainty in Semantic Web Taxonomies**

Chapter 10 presented a graph notation for representing uncertainty and conceptual overlap in Semantic Web taxonomies, and a Bayesian method for computing degrees of overlap between any two concepts of such a taxonomy. This method could be then utilized in the ranking of search results in *PFSS*. This method was later further developed for modeling geo-spatial changes over time [63].

2. **Creating User-centric Facets for Search**

The *PFSS* framework presented in Chapter 11 contains a facility to define search facets separately from annotation ontologies and probabilistically map the former to the latter. Boolean combinations of annotation concepts can be used in the mappings. *PFSS* defines clear probabilistic semantics for these facets, the mappings, and the Boolean combinations of concepts. Appendix A presents an implementation of this formalism using *RDF*.

3. **Ranking of Search Results**

The *PFSS* framework, provides efficient ranking of documents based on the

probability of relevance of each document computed based on weighted annotations, ontology structure, and mappings of end-user search facets to annotation ontologies. Clear probabilistic semantics are defined to all of these constructs.

4. **Combining Evidence of Document Relevance from Multiple Ranking Schemes**

PFSS supports the probabilistic combination of evidence from multiple ranking schemes to reach the final probability of relevance of a document.

5. **An Implementation of *PFSS***

Appendix A describes the technical design of a complete reference implementation of *PFSS*.

12.2 **Lessons Learned**

The probabilistic approach proved to be appropriate for solving the problems of *CFSS* summarized in Chapter 4, and the deficiencies noticed with the fuzzy approach. Bayesian probability theory provides a good framework for reasoning under uncertainty, because the agent's rational degrees of belief follow the rules of probability [111, 29]. Probability theory provides also good mechanisms for evidence combination.

The method for modeling uncertainty in Semantic Web taxonomies provides a simple extension to Semantic Web ontology languages by which conceptual overlap between concepts can be computed. This extension is simple and can be represented in RDF easily. Using the notation does not require deep knowledge of probability or set theory. The concepts can be quantified automatically, based on data records annotated according to the ontology, for example.

The notation of *PFSS* for representing the search categories, their mappings to annotation ontologies that might use Boolean combinations of concepts, and weighted annotations are also implemented using RDF, and can be easily used without extensive knowledge of probability theory. The use of these constructs is not mandatory for *PFSS*. In fact, a probabilistic faceted semantic search application can be created from any ontological knowledge base. In this case, of course, the more sophisticated features of *PFSS*, such as the Boolean combinations of concepts, separate end-user facet definitions etc. would not be used. However, usable rankings would still be provided.

Although substantial amount of work around probabilistic search has been done [88, 20, 104, 91, 24], a probabilistic framework for faceted search and faceted semantic search was missing. This part presented such a framework. Compared to previous probabilistic information retrieval systems, some of which use Bayesian networks such as the *Inference Network Model for Information Retrieval* [104] and the *Belief Network Model* [91], the *PFSS* has the following benefits: First, in *PFSS*, the content of documents and queries are modeled using ontological concepts. This captures the semantics of the documents better than the index-term based *bag-of-words* representation in [104, 91]. For example, the *homonym* and *synonym* problems that plague the term-based information retrieval systems are mostly non-existent in concept-based systems. Second, in *PFSS* we also model the relationships between the concepts, which allows for more realistic modeling of the documents and the searches than was possible in the previous approaches, where the words, or topics [24], are modeled as conditionally independent or mutually exclusive quantities. Third, we utilize semantics representable in an ontological notation such as RDF, to create conceptually complex search categories that can be composed for example as Boolean combinations of more simple search categories. Recall that Boolean queries are shown to be highly effective, but—due to the relative difficulty of their definition—underused types of queries [43]. *PFSS* solves the problem by defining these queries beforehand, and presenting them to the user in an easily graspable

manner in the search GUI. The previous approaches still are only prepared to respond to free-term searches entered by the user. The next part will present evidence based on empirical evaluation using realistic data that the framework provides good search results for real-world data.

Part IV

EVALUATION AND CONCLUSIONS

13 Empirical Comparison of Ranking Methods

This chapter presents a comparative evaluation of the *PFSS*, *FFSS*, *CFSS*, and a heuristic ranking method currently used in the *HealthFinland*. The evaluation is conducted using a real-world dataset taken from the *HealthFinland* [60] information portal.

In the evaluation we compare the following ranking methods:

1. **Crisp Faceted Semantic Search (CFSS)**

The method presented in Chapters 2–4. *CFSS* is considered the baseline in this evaluation, because it still is the most widely used faceted semantic search method. The evaluation presented in this chapter is in fact the first one to be published.

2. **Heuristic Ranking Method (HRM)**

The heuristic ranking method experimentally used in the *HealthFinland* portal.

3. **Fuzzy Faceted Semantic Search (FFSS)**

The method presented in Chapters 5–8.

4. **Probabilistic Faceted Semantic Search (PFSS)**

The method presented in Chapters 9–12.

In addition to the comparison of the above ranking methods we evaluate also the effects of combining ranking schemes. Recall that *PFSS* contains a facility to combine multiple ranking schemes. We will use the above ranking methods as individual ranking schemes, combine them pairwise, into groups of three, and finally combining

all four schemes. The ranking performance of each combination will be evaluated, to learn how combining ranking schemes influences ranking performance.

The rest of the chapter is organized as follows: First, the *HealthFinland* health portal is briefly introduced. Then the dataset used in the evaluation, which is taken from the *HealthFinland* portal is outlined. After this we describe, how the dataset was interpreted in each of the compared ranking methods. And finally, we present the results of the evaluation. We evaluate the methods using a statistical comparison of ranking results.

The next chapter (14) will summarize and discuss the results of this dissertation work.

13.1 HealthFinland — A Semantic Health Portal for the General Public

HealthFinland is a prototype of a national semantic health information portal created for the general public. It utilizes the faceted semantic search paradigm and it provides citizens with intelligent searching and browsing services to reliable and up-to-date health information created by various health organizations in Finland. The system is based on a shared semantic metadata schema and ontologies. The content includes metadata of thousands of Web documents such as webpages, articles, reports, campaign information, news, services, and other information related to health. The ontologies used in *HealthFinland* include the following:

1. FinMeSH²⁵—the Finnish translation of MeSH²⁶ by Duodecim²⁷. FinMeSH is

²⁵<http://194.89.160.67/codeserverTES/distribution-action.do?action=find&type=1&key=1172>

²⁶<http://www.nlm.nih.gov/mesh/>

²⁷<http://www.duodecim.fi>

used in *SKOS/RDF* format.

2. The national Finnish upper-ontology (YSO)²⁸ [56]. YSO is used in *RDF/OWL* format.
3. The European Multilingual Health Thesaurus (HPMULTI)²⁹. *HPMULTI* is used in *SKOS/RDF* format.

A major idea of *HealthFinland* is to provide the end-user with intelligent services for finding the right information based on his/her own conceptual view to health, and for browsing the contents based on their semantic relations. The views and vocabularies used in the end-user interface may be independent of the content providers organizational perspective, and are based on a *layman's* vocabulary that is different from the medical expert vocabularies used by the content providers in indexing the content. This *layman's* vocabulary has been created using a card sorting method to elicit how users tacitly group and organize concepts in the health domain. These end-user search facets are then mapped to the ontologies used in annotation using the properties from the SKOS Extension ontology [11]. [60]

13.2 The Dataset

As a dataset we used the following sample from the *HealthFinland* portal contents:

1. The *HealthFinland* annotation ontologies (FinMesh, YSO, HPMULTI) which are mapped onto each other in the project to form in essence one large crisp concept taxonomy.
2. The end-user facets of *HealthFinland* with their mappings to the annotation ontologies.

²⁸<http://www.seco.tkk.fi/ontologies/ys/>

²⁹<http://www.hpmulti.net/>

3. The *Nettineuvola*³⁰ repository of 133 documents containing pregnancy and infant support information. This document set was created as part of a regional Finnish health project, entitled *Healthy Kuopio*³¹. In order to include this document repository in the *HealthFinland* portal, the documents were annotated according to the *HealthFinland* ontologies by an information scientist using the semantic metadata schema [102] of *HealthFinland*. The annotation of the document set took about 2 working days. The annotations made by the information scientist were crisp. As part of the evaluation they were weighted automatically, as described in Section 13.3.1.

13.3 Interpreting the Data

This section explains the principles by which the data was interpreted and processed by the compared ranking methods. First the annotations and annotation concepts will be described and then the end-user facet definitions.

13.3.1 Annotations and Annotation Concepts

The property used for annotating the documents with annotation ontology concepts was *healthFinland : subject* [102]. This metadata property is intended to be used to describe the subject matter of the annotated document.

The document relevances in relation to the annotation concepts were computed in the following way for the different compared ranking methods:

CFSS The relevance is either 1.0, if the document is annotated to the annotation

³⁰<http://www.nettineuvola.fi>

³¹<http://www.tervekuopio.fi/>

concept or any of the concepts in the transitive closure of the subconcept relationship, or 0 otherwise.

HRM The annotation weight was computed using the formula: $W_{D,C} = 1/\sqrt{N}$ where $W_{D,C}$ is the annotation weight of the document D in relation to concept C and N is the total number of annotations of D . Annotation concept relevance was computed based on semantic distance, such that in the subconcept hierarchy passing an arrow reduces the relevance by a factor of 0.8. For example, if Di is annotated to concept A with weight 1.0, and A is a direct subconcept of B , and B is a direct subconcept of C , then the relevance of Di to C is $1.0*0.8*0.8 = 0.64$. In the case of multiple relevances for a document specific to one annotation concept—resulting, e.g. from multiple inheritance or multiple annotations—the maximum relevance value is chosen.

FFSS Annotation concept specific relevance—i.e., degree of membership of the document in the fuzzy set modeling the annotation concept—was computed according to the *FFSS* method described in Chapter 6. Annotation weights were determined similarly as in *HRM* above.

PFSS The annotation weight was determined similarly as in the above two methods.

13.3.2 End-user Facets

The end-user search categories which constitute the end-user facets were represented as instances of the *healthFinland:Category* class. The category hierarchies were represented using the *skos:broader* property.

The end-user categories were mapped to the annotation concepts using two properties. These were *skosext:narrowMatch* and *skosext:exactMatch*. These properties were used in the compared ranking methods as follows:

CFSS The mappings were interpreted in the same way as the subconcept relationship between annotation concepts above. Thus, a document matches the search category, if it matches the mapped annotation concept.

HRM In *HRM* the mapping properties were interpreted as follows:

- *skos:narrowMatch* was interpreted in the same way as the subconcept relationship in the annotation ontologies. In effect, if a document matched an annotation concept with weight 1, and this annotation concept was mapped to a search category with narrow match then the document matched the search category with weight 0.8.
- *skos:exactMatch* In exact match the weight of the mapping was 1.0, i.e., the relevance of the document was not reduced by *skos:exactMatch*.

FFSS Following the intuition presented above for *HRM*, for which these properties were initially defined, in *FFSS* the properties were interpreted as follows:

- *skosext:narrowMatch* was interpreted as fuzzy subsumption with fuzziness value of 0.8.
- *skosext:exactmatch* was taken as implication between fuzzy sets.

PFSS Following the intuition of the above method, in *PFSS* the properties were interpreted as follows:

- *skosext:narrowMatch* was interpreted as a search category mapping with weight 0.8.
- *skosext:exactmatch* was interpreted as a search category mapping with weight 1.0.

Using this information a *HealthFinland* specific facet mapper was implemented which precomputed the relevance of each document in relation to each end-user search category.

In this dataset all the categories were mapped to simple annotation concepts, and Boolean concepts were not used. This, of course, leaves much of the power of *FFSS*, and *PFSS* unused.

13.4 Evaluation

For the evaluation test we chose 10 search categories from the dataset described in the previous subsection. We used each of these search categories as a separate search, and asked a subject domain expert to define the relevant documents for each search. These categorizations made by the subject domain expert are called the *gold standard* in the following text. The search categories were chosen mainly based on the amount of matching documents (between 10 and 60 matches for each).

To compare the quality of the rankings produced by the different individual methods and the combinations statistically, we compared the ranking methods pairwise using Bayesian data analysis for multinomial data [38] as follows:

1. For each of the ranking methods we counted the number of errors made in the ranking of each document. A ranking method received an error point for a document ranking if one of the following was true:
 - A document was ranked by the studied ranking method among the R most relevant documents for a search category—where R is the total number of relevant documents for the current search according to the gold standard [20]—but this document was not among the relevant documents according to the gold standard.
 - A document was not ranked by the studied ranking method among the R most relevant documents for a search category, but this document was among the relevant documents according to the gold standard.

2. Then for the compared pair of ranking methods A and B we classified each document Di in one of the following classes:
 - Di^A , if the the ranking method A produced less errors in the ranking of the document Di than ranking method B .
 - Di^B , if the ranking method B produced less errors in the ranking of the document Di than ranking method A .
 - Di^0 , if the amount of errors for the document Di was equal according to both ranking methods.

Thus, for each compared pair of ranking methods the evaluation consisted of $n(D) = 133$ units of test data, such that $(n(Di^A), n(Di^B), n(Di^0))$ follows the multinomial distribution with parameters $(\theta_A, \theta_B, \theta_0)$, the proportions of the values of the above classification. Our estimand of interest here is $\theta_A - \theta_B$ the difference in the proportion of classifications where A performed better when compared to B .

3. We set a non informative prior distribution on θ , $\alpha_A = \alpha_B = \alpha_0 = 0$. The posterior distribution for $(\theta_A, \theta_B, \theta_0)$ is *Dirichlet* $(n(Di^A), n(Di^B), n(Di^0))$.
4. Then—to estimate the statistical significance of the difference in the classification performance—we sample 1000 points $(\theta_A, \theta_B, \theta_0)$ from the posterior Dirichlet distribution and compute $\theta_A - \theta_B$ for each point. The number 1000 of samples was chosen because it is a rather standard amount of samples in Bayesian data analysis.
5. Our confidence that ranking method A produces better ranking results than B is $n(\theta_A - \theta_B > 0)/1000$, where $n(\theta_A - \theta_B > 0)$ is the number of points where $\theta_A > \theta_B$.

The results of this evaluation are presented in Table 13.1. The table present the statistically at least marginally significant results. To visualize the difference in

ranking performance between the individual ranking methods, Figure 13.1 presents the R-precision values averaged over the 10 compared searches for the four compared algorithms. Recall, that R-precision evaluation generates a single-value summary of the ranking by computing the precision at the R^{th} position in the ranking, where R is the total number of relevant documents for the current search according to the gold standard [20].

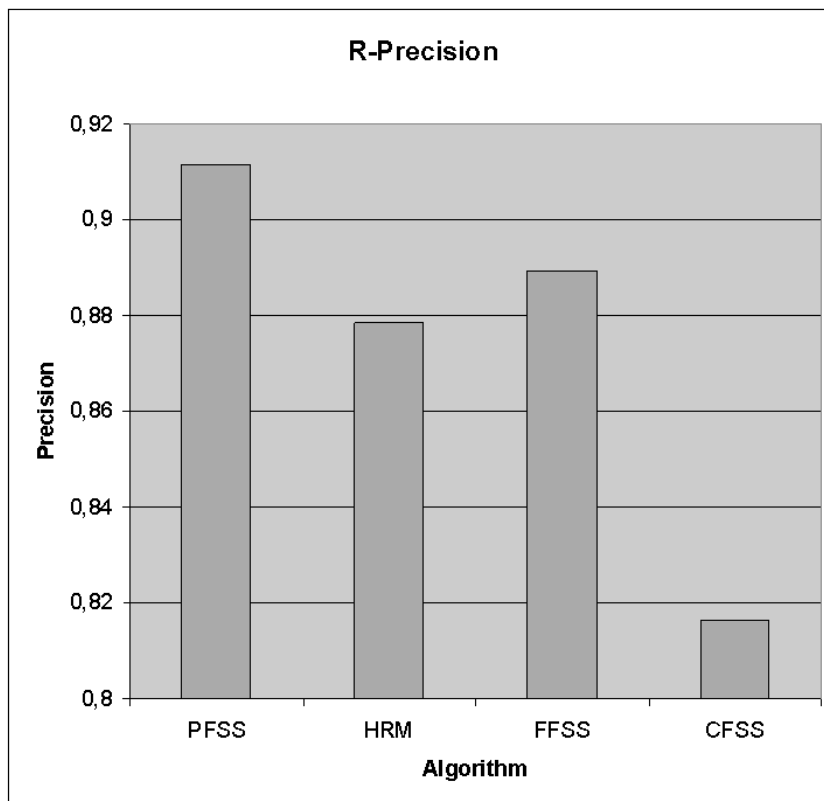


Figure 13.1: The average R-Precision values for the four algorithms compared.

Based on the comparison of the individual ranking methods presented in Table 13.1 and visualized in Figure 13.1, the following observations can be made:

1. Both *PFSS*, and *FFSS* significantly outperform the baseline *CFSS*.
2. *PFSS* outperforms *HRM*. This result is marginally significant, i.e.

Table 13.1: Results of the Bayesian Statistical Comparison of Ranking Methods.

A	B	Dirichlet(A,B,0)	$P(A > B)$
PFSS	CFSS	(46,20,71)	0.999
PFSS	HRM	(31,20,82)	0.942
PFSS	CFSS&HRM	(34,19,80)	0.983
FFSS	CFSS	(44,17,79)	0.999
FFSS	CFSS&HRM	(34,21,78)	0.963
HRM	CFSS	(38,22,73)	0.987
PFSS&FFSS	CFSS	(45,18,70)	0.999
PFSS&FFSS	HRM	(31,19,83)	0.961
PFSS&FFSS	CFSS&HRM	(31,16,86)	0.980
PFSS&HRM	CFSS	(42,20,71)	0.999
PFSS&HRM	CFSS&HRM	(29,18,86)	0.942
PFSS&CFSS	CFSS	(42,20,71)	0.999
PFSS&CFSS	CFSS&HRM	(31,20,82)	0.942
FFSS&HRM	CFSS	(43,15,75)	0.999
FFSS&HRM	HRM	(33,20,830)	0.962
FFSS&HRM	CFSS&HRM	(32,14,87)	0.993
CFSS&HRM	CFSS	(48,35,50)	0.917
FFSS&CFSS	CFSS	(42,19,71)	0.999
PFSS&FFSS&HRM	CFSS	(45,20,68)	0.999
PFSS&FFSS&HRM	CFSS&HRM	(27,14,92)	0.985
PFSS&FFSS&CFSS	CFSS	(45,18,70)	0.999
PFSS&FFSS&CFSS	HRM	(31,19,83)	0.961
PFSS&FFSS&CFSS	CFSS&HRM	(31,16,86)	0.980
PFSS&CFSS&HRM	CFSS	(42,20,71)	0.997
PFSS&CFSS&HRM	CFSS&HRM	(29,18,86)	0.942
FFSS&CFSS&HRM	CFSS	(43,15,75)	0.999
FFSS&CFSS&HRM	HRM	(33,20,80)	0.962
FFSS&CFSS&HRM	CFSS&HRM	(32,14,87)	0.993
FFSS&CFSS&HRM	FFSS&CFSS	(10,2,121)	0.990
PFSS&FFSS&CFSS&HRM	CFSS	(45,19,69)	0.999
PFSS&FFSS&CFSS&HRM	HRM	(31,19,83)	0.961
PFSS&FFSS&CFSS&HRM	CFSS&HRM	(29,14,90)	0.992
PFSS&FFSS&CFSS&HRM	PFSS&CFSS&HRM	(6,2,125)	0.928

$$P(PFSS > HRM) = 0.942 > 0.9.$$

3. *FFSS* shows tendency to outperform *HRM*—as shown in Figure 13.1—but this tendency did not reach statistical significance, i.e., $P(FFSS > HRM) = 0.872$, which is a little bit below the limit of marginal significance.
4. According to Figure 13.1 *PFSS* seems to outperform *FFSS*, however the difference is not statistically significant $P(PFSS > FFSS) = 0.628$.

Notice, that the more advanced features of *PFSS* and *FFSS*, such as mappings of search categories to Boolean combinations of annotation concepts were not used. This probably handicapped the ranking results of *PFSS*, and *FFSS* when compared to *HRM* and *CFSS*.

Based on the evaluation, the following observations can be made regarding the effects of combining ranking schemes:

1. At each level of combinations—i.e., individuals, pairs, triples, all four schemes—the worst combination was outperformed by the worst on the the next level. Specifically, The worst individual ranking method *CFSS* was outperformed by the worst pairwise combination of ranking schemes *CFSS*⊗*HRM*. *CFSS*⊗*HRM* was then outperformed by the worst combination of three schemes *PFSS*⊗*CFSS*⊗*HRM*, which was outperformed by the combination of all four ranking methods *PFSS*⊗*FFSS*⊗*CFSS*⊗*HRM*.
2. There were no significant differences between the ranking performances of the best combinations of each layer.
3. In terms of number of outperformed combinations, the best combination of each layer outperformed at least as many combinations as the best combination on a level directly beneath it. Specifically, the best individual ranking

method *PFSS* outperformed three combinations. The best pairwise combination *FFSS*⊗*HRM* also outperformed three combinations. The best combination of three schemes outperformed four combinations, as did the combination of all four schemes.

4. The combinations *PFSS*⊗*CFSS*, and *FFSS*⊗*CFSS* significantly outperformed *CFSS*, but neither *PFSS* nor *FFSS* outperformed either of these combinations.
5. Interestingly, in some situations it seemed that adding a bad ranking scheme to a combination improved rather than impaired the ranking performance. For example, *FFSS*⊗*CFSS*⊗*HRM* outperformed more combinations than *FFSS*⊗*HRM*. In this sense, the combination seems in some situations to be more than the sum of its parts, possibly because the different ranking schemes eliminate the errors of each other.
6. There were no cases in which a combination performed significantly worse than any of its components.

As the overall conclusion of the above points regarding the effects of combining ranking schemes, it could be said that on average combining ranking schemes has a positive effect on ranking performance. It seems that the combination usually performs more like the best individual scheme included rather than the worst.

14 Results and Discussion

This chapter presents a concluding summary of this dissertation. It contains a compilation of the results of this work, and summarizes the discussion of these results.

14.1 Results

The central results presented in this dissertation are the following:

1. **Rule-based Projection of Facets from Annotation Ontologies**

Chapter 3 presented a method and an algorithm to create facets algorithmically from Semantic Web ontologies, using logical rules implemented in *SWI-Prolog* [13].

2. **Rule-based Creation of Recommendation Links based on Ontology Structure**

Chapter 3 also contained a method to create recommendation links for search items. As in the case of facet projection, these recommendations were created using logical rules based on the annotation ontologies, and were implemented as *SWI-Prolog* predicates.

3. **Semantic Web HTML Generator (SWeHG) Tool**

SWeHG is a tool to create static HTML websites from *semantic knowledge bases*. Sites created with SWeHG consist from the following main ingredients: First, content pages, entitled *resource pages*, that are linked to other resources pages by recommendation links using the above rule-based recommendation link creation method. Second, indices to the resource pages that are projected

from the annotation ontologies using the rule-based facet projection method presented. Thus, a website created with *SWeHG* is a *single-faceted* search application.

4. **Modeling of Annotation Related Uncertainty**

Chapter 6 introduced the notion of weighted annotations. The annotation weight is a real number in the range $[0, 1]$, and it represents the degree of relevance of a document or search item to the concept that it is annotated to. Weighted annotations are utilized both in *FFSS* and *PFSS* to provide ranking of search results.

5. **Ontological Extension to *TF-IDF* for Weighting Document Annotations**

Chapter 7 presented a method by which crisp annotations can be algorithmically weighted. This method is an ontological extension of the widely used *TF-IDF* term weighting method.

6. **Fuzzy Set based Method for Ranking Faceted Semantic Search Results**

In Chapter 6, we extended the set theoretic basis of faceted search from crisp to fuzzy sets, such that a search item can have degrees of membership in search categories instead of either belonging or not in these categories. This degree of membership is then used as a criterion for ranking search results. The weighted annotations are used to represent the degree of membership of search items in annotation concepts.

7. **Fuzzy Set based Method for Mapping Separate End-user Facets to Annotation Ontologies**

Chapter 6 also contained a fuzzy set based method to map separate end-user facets to annotation ontologies. In this method search facets can be fuzzily mapped to annotation concepts, and this mapping is interpreted as fuzzy implication—i.e., fuzzy inclusion—of the fuzzy set corresponding to the

annotation concept in the fuzzy set that corresponds to the search category. In addition to simple annotation concepts, search categories can be mapped also to Boolean combinations of annotation concepts.

8. **Fuzzy Faceted Semantic Search (FFSS) Framework**

FFSS combines Results 4–7 to accomplish a fuzzy set based extension to crisp faceted semantic search. This framework provides mapping of separate end-user facets to annotation ontologies, and ranking of search results. The framework includes the required formalisms to represent weighted annotations, taxonomies and Boolean combinations of annotation concepts, hierarchies of search categories, and mappings of search categories onto annotation concepts. These representations are provided with a fuzzy set interpretation. The framework also provides the algorithms to compute the ranked result sets for faceted search queries.

9. **Probabilistic Method for Modeling Uncertainty in Semantic Web Taxonomies**

Chapter 10 presented a graph notation for representing uncertainty based on conceptual overlap in Semantic Web taxonomies, and a probabilistic method for computing degrees of overlap between the concepts of such a taxonomy.

10. **Probabilistic Method for Ranking Faceted Semantic Search Results**

Section 11.3 presented a probabilistic method to compute the probabilities of search item relevance to an annotation concept, and how this probability can be used to compute search category specific probability of relevance. The method incorporates the probabilistic model of Chapter 10 originally developed for modeling uncertainty in Semantic Web taxonomies.

11. **Probabilistic Method for Combining Evidence for Ranking of Search Results**

Section 11.6 presented an extension to the above probabilistic method to rank search result, which enables combination probabilities of search item

relevance computed using multiple ranking schemes.

12. **Probabilistic Method for Mapping Separate End-user Facets to Annotation Ontologies**

Chapter 11 presents various ways to map separately created end-user facets to annotation ontologies. Search categories can be mapped onto simple annotation concepts as well as Boolean combinations of concepts. A probabilistic interpretation for these mappings was developed.

13. **Probabilistic Faceted Semantic Search (PFSS) Framework**

Results 9–12 above were compiled to create the *PFSS* framework. Thus, *PFSS* provides sophisticated mapping of separate end-user facets onto annotation ontologies, ranking of search results, and combination of evidence from multiple ranking schemes to compute probabilities of search item relevance.

14. **Probabilistic Faceted Semantic Search Tool**

A complete implementation of *PFSS* is presented in appendix A. The implementation contains an *OWL* schema to be used in the implementation of the semantic knowledge base, a Jena-based implementation of the semantic knowledge base, and a Java Spring³² implementation of the required algorithms for the model creation and the computation of the document relevances.

15. **Comparative Evaluation of *CFSS*, *FFSS* and *PFSS***

Chapter 13 presented an extensive evaluation of quality of rankings produced by *PFSS* and *FFSS*, to *CFSS* and *HRM*. As a summary *PFSS* proved to produce best ranking quality, but also *FFSS* significantly outperformed *CFSS* and showed tendency to outperform also *HRM*.

³²<http://www.springsource.org/>

16. Comparative Evaluation of the Effects of Combining Ranking Schemes

Chapter 13 presented an extensive evaluation of the effects of combining ranking schemes. According to the evaluation combining ranking has a positive effect on ranking, and it seems that the combination usually performs more like the best individual scheme.

14.2 Discussion

The rule-based facet projection and recommendation link generation solutions developed for *SWeHG* [49] were later utilized as part of the recommendation and facet projection engine of *OntoViews*, which is the faceted semantic search tool developed by the *Semantic Computing (SeCo)* research group³³ [108, 72]. This technology is the basis of such *FSS* applications as *MuseumFinland* [52], *Orava* [65], *HealthFinland* [60], *Veturi* [81], and *SW-Suomi.fi* [95]. Although the algorithmic projection of facets from ontologies was already implemented before *SWeHG* in the *Promoottori* [57] application, the rule-based projection, which enables more flexible and intelligent definition of facets and recommendations was first developed for *SWeHG*. The *SWeHG* tool was also important in realizing the limitations of the *CFSS* paradigm, namely, the inability to model uncertainty, the inability to rank search results according to relevance, and the usability problems resulting from presenting annotation ontology concepts as search categories.

The *FFSS* framework provided a solution to the problems of *CFSS*, essentially, by extending the crisp set basis of *FSS* to fuzzy sets. *FFSS* proved to be a rather straight forward framework to design and implement. According to evaluation presented in Chapter 13 this method provides good search results. The *FFSS* framework did get some inspiration from fuzzy versions of description logics, such as [97].

³³<http://www.seco.tkk.fi/>

However, in *FFSS* we concentrated on faceted search, which according to our knowledge, has not been extended to fuzzy sets before. However, the fuzzy set approach to uncertainty has been criticized because of its heuristic nature when compared to probability theory [26].

The ontological extension to *TF-IDF* by which annotations can be weighted is a partial solution to the lack of uncertainty modeling capabilities of Semantic Web ontologies. It provides a way to algorithmically weight hand-made crisp annotations, which increases the applicability of both *FFSS* and *PFSS*. Because the weights created by the extended *TF-IDF* method fall in the range $[0, 1]$ the weights can be interpreted as probabilities of relevance as well, so this method can be used also in weighting annotations for *PFSS*. The ontological extension to *TF-IDF* provides also some benefits when compared to traditional *TF-IDF* weighting: First, terms that are expressions of the same concept can be represented using a single concept identifier which results in a compressed document representation, and second the concept hierarchies of the ontologies can be utilized to enable better query answering.

We developed also a Bayesian probabilistic extension to *CFSS*, namely the *PFSS* framework. As opposite to the heuristic nature of the fuzzy approach, it has been shown that agent's rational degrees of belief follow the rules of probability. Thus, probability theory is more than a heuristic for uncertainty modeling, but the only approach that is proved to answer for consistency requirements for reasoning under uncertainty [29, 111]. This also means, that fuzzy logic, which has a different set of axioms, necessarily violates these consistency requirements. In addition, combination of evidence from multiple sources is at the core of probabilistic reasoning.

When compared to probabilistic models for information retrieval found in the literature [88, 20, 104, 91, 24], *PFSS* provides the following benefits: First, none of the earlier models is developed for, or supports, faceted search. For example, they lack the means to represent search category hierarchies, or methods to compute rank-

ing over a combination of search category selections. Second, in *PFSS* the content of documents and searches are modeled using ontological concepts, whereas previous models, such as [104, 91], represent documents and queries as *bags-of-words*. Ontological concept representation is more compressed, and also usually solves problems of synonyms and homonyms encountered in term-based information retrieval. Third, *PFSS* also models the relationships between concepts, which allows for more realistic representation of the documents than is possible in approaches found in the literature. The usual explanation for the lack of modeling of the relationships between terms is the computational cost of such a modeling. However, the faceted search paradigm allows us to precompute all search category specific probabilities, which largely solves the potential performance problems of sophisticated probability models. Fourth, *PFSS* utilizes semantics representable in an ontological notation to create conceptually sophisticated search categories, that can be composed from Boolean combinations of more simple search category definitions. This means, that *PFSS* enables us to offer the user sophisticated Boolean queries, packaged and served as intuitive search categories. Recall, that Boolean queries have been shown to be highly effective queries that are underused, due to the fact that their construction requires substantial amount of expertize in search [43]. Other probabilistic models still are only prepared to respond to free-term searches entered by the user.

The methods developed in this thesis require a fair amount of human modeling and knowledge engineering work, which could be seen as a possible limitation of these methods. A large parts of the methods are devoted to handling Boolean combinations of concepts. These are still rather rarely used in real-world Semantic Web ontologies. This is manifested in the empirical evaluation conducted for the thesis: A dataset that utilizes Boolean concepts in the definition of search categories was not available. This shortcoming most probably handicapped the performance of *FFSS* and *PFSS* in comparison to the heuristic ranking method. Due to this deficiency, the rather small size of the used dataset, and the fact that the gold standard was constructed on based on relevance evaluation of a single person, the

empirical evaluation still has to be considered preliminary.

In the future, we would like to develop *PFSS* further by adding a mechanism to learn and improve ranking performance based on user feedback, or other data gathered from the usage of the system.

References

- [1] Dublin Core Metadata Element Set, Version 1.1. <http://dublin-core.org/documents/dces/>.
- [2] International Statistical Classification of Diseases and Related Health Problems, 10th Revision (ICD-10), Version for 2007. <http://www.who.int/classifications/apps/icd/icd10online/>.
- [3] Jena Framework. <http://jena.sourceforge.net/javadoc/index.html>.
- [4] Medical Subject Headings. <http://www.nlm.nih.gov/mesh/meshhome.html>.
- [5] Medical Subject Headings (MeSH/FinMeSH). <http://www.seco.tkk.fi/ontologies/mesh/>.
- [6] OWL Web Ontology Language Guide. <http://www.w3.org/TR/2003/CR-owl-guide-20030818/>.
- [7] RDF Primer. <http://www.w3.org/TR/rdf-primer>.
- [8] RDF Vocabulary Description Language 1.0: RDF Schema. <http://www.w3.org/TR/rdf-schema/>.
- [9] RDF/XML Syntax Specification (Revised). <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [10] SKOS Core Guide. <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/>.
- [11] SKOS Extensions Vocabulary Specification. <http://www.w3.org/2004/02/-skos/extensions/spec/>.
- [12] SKOS Mapping Vocabulary Specification. <http://www.w3.org/2004/02/-skos/mapping/spec/>.

- [13] SWI-Prolog documentation. <http://www.swi-prolog.org/pldoc/index.html>.
- [14] Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martínez. 2000. Enriching very large ontologies using the WWW. In: in Proceedings of the Ontology Learning Workshop. ceur-ws.
- [15] J. Aitchison, A. Gilchrist, and D. Bawden. 2000. Thesaurus Construction and Use: A Practical Manual. London.
- [16] Giorgos Akrivas, Manolis Wallace, Giorgos Andreou, Giorgos Stamou, and Stefanos Kollias. 2002. Context-Sensitive Semantic Query Expansion. In: ICAIS '02: Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02), page 109. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-1733-1.
- [17] Enrique Alfonseca and Suresh Manandhar. 2002. Extending a Lexical Ontology by a Combination of Distributional Semantics Signatures. In: EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pages 1–7. Springer-Verlag, London, UK. ISBN 3-540-44268-5.
- [18] R.A. Angryk and F.E. Petry. 2003. Consistent fuzzy concept hierarchies for attribute generalization. In: Proceeding of the IASTED International Conference on Information and Knowledge Sharing (IKS' 03). ACTA Press, Scottsdale, AZ, USA. ISBN 0-88986-396-2.
- [19] Nathalie Aussenac-Gilles and Dagobert Sörgel. 2005. Text analysis for ontology and terminology engineering. *Appl. Ontol.* 1, no. 1, pages 35–46.
- [20] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. Modern Information Retrieval. Addison Wesley.
- [21] Sean Bechhofer, Leslie Carr, Carole Goble, and Wendy Hall. 2001. Conceptual Open Hypermedia = The Semantic Web? In: Proceedings of the WWW2001, Semantic Web Workshop, Hongkong, pages 44–50.

- [22] K. Beck. 1999. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.
- [23] T. Berners-Lee, J. Handler, and O. Lassila. 2001. The Semantic Web. *Scientific American* 284, no. 5, pages 34–43.
- [24] David M. Blei, Michael I. Jordan, and Andrew Y. Ng. 2003. Hierarchical Bayesian Models for Applications in Information Retrieval. In: *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*. Oxford University Press.
- [25] G. Bordogna, P. Bosc, and G. Pasi. 1996. Fuzzy inclusion in database and information retrieval query interpretation. In: *SAC '96: Proceedings of the 1996 ACM symposium on Applied Computing*, pages 547–551. ACM, New York, NY, USA. ISBN 0-89791-820-7.
- [26] P. Cheeseman. 1986. Probabilistic versus Fuzzy Reasoning. In: L. N. Kanal and J. F. Lemmer (editors), *Uncertainty in Artificial Intelligence*, pages 85–102. Elsevier Science Publishers B.V. (North-Holland).
- [27] G. Chen and T. T. Pham. 2001. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press.
- [28] Philipp Cimiano and Steffen Staab. 2005. Learning Concept Hierarchies from Text with a Guided Hierarchical Clustering Algorithm. In: *In Proceedings of the Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*.
- [29] R. T. Cox. 1979. Of Inference and Inquiry - An Essay in Inductive Logic. In: Levine and Tribus (editors), *The Maximum Entropy Formalism*, pages 119–167. MIT Press.
- [30] A. Dellit and T. Boston. 2007. Relevance Ranking of Results from MARC-based Catalogues: From Guidelines to Implementation Exploiting Structured

- Metadata. In: Information Online 2007, 13th Exhibition and Conference. Sydney Convention and Exhibition Centre, Australia.
- [31] M. Dewey. 2008. A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library. The Echo Library.
- [32] Zhongli Ding and Yun Peng. 2004. A Probabilistic Extension to Ontology Language OWL. In: HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4, page 40111.1. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-2056-1.
- [33] Andreas Faatz and Ralf Steinmetz. 2002. Ontology Enrichment with Texts from the WWW. In: In Semantic Web Mining, WS02.
- [34] C. Fellbaum. WordNet: An Electronic Lexical Database.
- [35] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster. 2002. Weaving the Semantic Web. The MIT Press.
- [36] F. V. Finin and F. B. Finin. 2001. Bayesian Networks and Decision Graphs. Springer-Verlag.
- [37] A. Gangemi, N. Guarino, C Masolo, A. Oltramari, and L. Schneider. 2002. Sweetening Ontologies with DOLCE. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. Springer-Verlag.
- [38] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. 2003. Bayesian Data Analysis. Chapman and Hall/CRC.
- [39] R. Giugno and T. Lukasiewicz. 2002. P-SHOQ(D): A Probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. INFSYS Research Report 1843-02-06, Technische Universität Wien.

- [40] A. Gómez-Pérez and D. Manzano-Macho. 2004. An overview of methods and tools for ontology learning from texts. *The Knowledge Engineering Review* 19, no. 03, pages 187–212. http://journals.cambridge.org/article_S0269888905000251.
- [41] T. Gu and D.Q. Zhang H.K. Pung. 2004. A Bayesian Approach for Dealing with Uncertain Contexts. In: *Proceedings of the 2nd International Conference on Pervasive Computing*. Austrian Computer Society.
- [42] Udo Hahn and Korné G. Markó. 2001. Joint knowledge capture for grammars and ontologies. In: *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 68–75. ACM, New York, NY, USA. ISBN 1-58113-380-4.
- [43] M. Hearst, A. Elliot, J. English, R. Sinha, K. Swearingen, and K.-P. Lee. 2002. Finding the flow in web site search. *CACM* 45, no. 9, pages 42–49.
- [44] M. Hildebrand, J. R. van Ossenbruggen, and L. Hardman. 2006. */facet: A browser for heterogeneous Semantic Web repositories*. In: *The Semantic Web - ISWC 2006*, pages 272–285. Springer-Verlag.
- [45] Markus Holi. 2004. *Modeling Uncertainty in Semantic Web Taxonomies*. Master of Science Thesis. Department of Computer Science, University of Helsinki, <http://ethesis.helsinki.fi/julkaisut/mat/tieto/pg/holi/>.
- [46] Markus Holi and Eero Hyvönen. 2006. *Fuzzy View-Based Semantic Search*. In: *Proceedings of the 1st Asian Semantic Web Conference (ASWC2006)*, Beijing, China. Springer-Verlag.
- [47] Markus Holi and Eero Hyvönen. 2006. *Modeling Uncertainty in Semantic Web Taxonomies*. In: Zhongmin Ma (editor), *Soft Computing in Ontologies and Semantic Web*. Springer-Verlag.

- [48] Markus Holi, Eero Hyvönen, and Petri Lindgren. 2006. Integrating tf-idf Weighting with Fuzzy View-Based Search. In: Proceedings of the ECAI Workshop on Text-Based Information Retrieval (TIR-06). CEUR-WS.
- [49] Eero Hyvönen, Markus Holi, and Kim Viljanen. 2004. Designing and Creating a Web Site Based on RDF Content. In: Proceedings of WWW2004 Workshop, Application Design, Development, and Implementation Issues.
- [50] Eero Hyvönen, Miikka Junnila, Suvi Kettula, Eetu Mäkelä, Samppa Saarela, Mirva Salminen, Ahti Syreeni, Arttu Valo, and Kim Viljanen. 2004. Finnish Museums on the Semantic Web. User's Perspective on MuseumFinland. In: Proceedings of Museums and the Web 2004 (MW2004).
- [51] Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Kim Viljanen Heini Kuittinen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio, and Katariina Nyberg. 2009. CultureSampo – A National Publication System of Cultural Heritage on the Semantic Web 2.0. In: Proceedings of the 6th European Semantic Web Conference (ESWC2009), Heraklion, Greece. Springer-Verlag.
- [52] Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. 2005. MuseumFinland – Finnish Museums on the Semantic Web. *Journal of Web Semantics* 3, no. 2, pages 224–241.
- [53] Eero Hyvönen, Samppa Saarela, and Kim Viljanen. 2003. Ontogator: Combining View- and Ontology-Based Search with Semantic Browsing. In: Proceedings of XML Finland 2003. Paper presented at the international SEPIA Conference, Helsinki, Sept. 18–20, 2003.
- [54] Eero Hyvönen, Samppa Saarela, and Kim Viljanen. 2004. Application of Ontology Techniques to View-Based Semantic Search and Browsing. In: The

- Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004).
- [55] Eero Hyvönen, Mirva Salminen, Suvi Kettula, and Miikka Junnila. 2004. A Content Creation Process for the Semantic Web. In: Proceedings of OntoLex 2004. Lisbon, Portugal.
- [56] Eero Hyvönen, Katri Seppälä, Kim Viljanen, and Matias Frosterus. 2007. Yleinen suomalainen ontologia YSO — kohti suomalaista semanttista webiä (General Finnish Ontology YSO—Towards the Finnish Semantic Web). In: Tietolinja.
- [57] Eero Hyvönen, Avril Styrman, and Samppa Saarela. 2002. Ontology-Based Image Retrieval. In: Towards the semantic web and web services, Proceedings of XML Finland 2002 Conference, pages 15–27.
- [58] Eero Hyvönen, Arttu Valo, Kim Viljanen, and Markus Holi. 2003. Publishing Semantic Web Content as Semantically Linked HTML Pages. In: Proceedings of XML Finland 2003, Kuopio, Finland. http://www.cs.helsinki.fi/u/eahyvone/publications/xmlfinland2003/swehg_article_xmlfi2003.pdf.
- [59] Eero Hyvönen, Kim Viljanen, Eetu Mäkelä, Tomi Kauppinen, Tuukka Ruotsalo, Onni Valkeapää, Katri Seppälä, Osma Suominen, Olli Alm, Robin Lindroos, Teppo Käsälä, Riikka Henriksson, Matias Frosterus, Jouni Tuominen, Reetta Sinkkilä, and Jussi Kurki. 2007. Elements of a National Semantic Web Infrastructure - Case Study Finland on the Semantic Web (Invited paper). In: Proceedings of the First International Semantic Computing Conference (IEEE ICSC 2007), Irvine, California. IEEE Press.
- [60] Eero Hyvönen, Kim Viljanen, and Osma Suominen. HealthFinland — Finnish Health Information on the Semantic Web. In: Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan , Korea.

- [61] Eero Hyvönen, Kim Viljanen, Osma Suominen, and Eija Hukka. 2008. HealthFinland – Publishing Health Promotion Information on the Semantic Web. In: Proceedings of DrMED 2008: International Workshop on Describing Medical Web Resources. The 21st International Congress on the European Federation for Medical Informatics (MIE 2008), Göteborg, Sweden.
- [62] Eero Hyvönen, Kim Viljanen, Jouni Tuominen, and Katri Seppälä. 2008. Building a National Semantic Web Ontology and Ontology Service Infrastructure—The FinnONTO Approach. In: Proceedings of the European Semantic Web Conference ESWC 2008. Springer.
- [63] Tomi Kauppinen and Eero Hyvönen. 2005. Modeling and Reasoning about Changes in Ontology Time Series. In: Rajiv Kishore, Ram Ramesh, and Raj Sharman (editors), *Ontologies in the Context of Information Systems*. Springer-Verlag. In press.
- [64] G. J. Klir and B. Yuan. 1995. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.
- [65] Teppo Känslä and Eero Hyvönen. 2006. A Semantic View-based Portal Utilizing Learning Object Metadata. 1st Asian Semantic Web Conference (ASWC2006), Semantic Web Applications and Tools Workshop.
- [66] Daphne Koller, Alon Levy, and Avi Pfeffer. 1997. P-CLASSIC: A tractable probabilistic description logic. In: In Proceedings of AAAI-97, pages 390–397. AAAI Press.
- [67] J. Kristensen. 1993. Expanding end-users’ query statements for free text searching with a search-aid thesaurus. *Information Processing and Management* 29, no. 6, pages 733–744.
- [68] J. Kristensen and K. Järvelin. 1990. The effectiveness of a searching thesaurus in free-text searching of a full-text database. *International Classification* 17, no. 2, pages 77–84.

- [69] Kenneth J. Laskey and Kathryn B. Laskey. 2008. Uncertainty Reasoning for the World Wide Web: Report on the URW3-XG Incubator Group. In: URSW.
- [70] Joon Ho Lee. 1997. Analyses of Multiple Evidence Combination. In: SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 267–276. ACM, New York, NY, USA. ISBN 0-89791-836-3.
- [71] R. L. Leskinen. 1997. Museoalan asiasanasto. Museovirasto, Helsinki, Finland.
- [72] Eetu Makelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen. 2004. On-toViews — A Tool for Creating Semantic Web Portals. In: Proceedings of the 3rd International Semantic Web Conference (ISWC 2004). Springer-Verlag.
- [73] A. Maple. 1995. Faceted access: a review of the literature. http://library.music.indiana.edu/tech_s/mla/facacc.rev.
- [74] G. Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Communications of the ACM* 49, no. 4, pages 41–46.
- [75] G. Marchionini and B. Brunk. 2003. Toward a General Relation Browser: A GUI for Information Architects. *Journal of Digital Information* 4, no. 1.
- [76] D. Mayrer and T. Warfel. 2003. Card Sorting: A Definitive Guide, Boxes and Arrows. <http://boxesandarrows.com/S1937>.
- [77] M. Mazzieri. 2004. A Fuzzy RDF Semantics to Represent Trust Metadata. In: 1st Workshop on Semantic Web Applications and Perspectives (SWAP2004), pages 83–89. Ancona, Italy.
- [78] M. Mazzieri and A. F. Dragoni. 2005. Fuzzy Semantics for Semantic Web Languages. In: Proceedings of the ISWC-2005 Workshop on Uncertainty Reasoning for the Semantic Web, pages 12–22.

- [79] Prasenjit Mitra, Natasha Noy, and Anuj Jaiswal. 2005. OMEN: A Probabilistic Ontology Mapping Tool. pages 537–547. Springer-Verlag, Berlin / Heidelberg. ISBN 978-3-540-29754-3.
- [80] Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff. 2005. View-Based User Interfaces for Information Retrieval on the Semantic Web. In: Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction.
- [81] Eetu Mäkelä, Kim Viljanen, Petri Lindgren, Mikko Laukkanen, and Eero Hyvönen. 2005. Semantic Yellow Page Service Discovery: The Veturi Portal. In: Poster paper, 4th International Semantic Web Conference.
- [82] Saikat Mukherjee, Guizhen Yang, and I. V. Ramakrishnan. 2003. Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis. In: In Intl. Semantic Web Conf. (ISWC, pages 533–549. Springer Berlin / Heidelberg.
- [83] R. E. Neapolitan. 2003. Learning Bayesian Networks. Prentice Hall.
- [84] R. T. Ng and V. S. Subrahmanian. 1993. A Semantical Framework for Supporting Subjective and Conditional Probabilities in Deductive Databases. Automated Reasoning Journal 10, no. 2, pages 191–235.
- [85] R. T. Ng and X. Tian. 1997. Semantics, Consistence and Query Processing of Empirical Deductive Databases. IEEE transactions on knowledge and data engineering 9, no. 1, pages 32–49.
- [86] A. S. Pollitt. 1998. The Key Role of Classification and Indexing in View-Based Searching. Technical report, University of Huddersfield, UK, <http://www.ifla.org/IV/ifla63/63polst.pdf>.
- [87] D. Polong, N. Henze, and W. Nejdl. 2003. Logic-based Open Hypermedia for the Semantic Web. In: Proceedings of the Int. Workshop on Hypermedia and the Semantic Web, Hypertext 2003 Conference. Nottingham, UK.

- [88] Tadeusz Radecki. 1988. Exploiting the Probability Ranking Principle to Increase the Effectiveness of Conventional Boolean Retrieval Systems. In: *Informetrics 87/88*, pages 209–218. Elsevier Science Publishers B. V.
- [89] S. R. Ranganathan. 1965. Colon Classification. Technical report, Graduate School of Library Service, Rutgers, the State University.
- [90] A. Rector. 2004. Defaults, Context, and Knowledge: Alternatives for OWL-Indexed Knowledge Bases. In: *Proceedings of Pacific Symposium on Bio-computing*, pages 226–237. World Scientific Publishing Co.
- [91] Berthier A. N. Ribeiro and Richard Muntz. 1996. A Belief Network Model for IR. In: *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260. ACM, New York, NY, USA. ISBN 0-89791-792-8.
- [92] G. Rugg and P. McGeorge. 1997. The Sorting Techniques: A Tutorial Paper on Card Sorts, Picture Sorts and Item Sorts. *Expert Systems* 14, no. 2, pages 80–97.
- [93] G. Salton and C. Buckley. 1987. Term Weighting Approaches in Automatic Text Retrieval. Technical report, Ithaca, NY, USA.
- [94] M. Schraefel, M. Karam, and S. Zhao. 2003. Listen to the Music: Audio Preview Cues for Exploration of Online Music. In: *Proceedings of Interact 2003 - Bringing the Bits Together*. IOS Press, Zürich, Switzerland.
- [95] Teemu Sidoroff and Eero Hyvönen. 2005. Semantic E-government Portals - A Case Study. In: *Proceedings of the ISWC-2005 Workshop on Semantic Web Case Studies and Best Practices for eBusiness SWCASE05*.
- [96] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks. 2005. The Fuzzy Description Logic f-SHIN. In: *Proceedings of the ISWC-2005 Workshop on Uncertainty Reasoning for the Semantic Web*.

- [97] Umberto Straccia. 2005. Towards a Fuzzy Description Logic for the Semantic Web (Preliminary Report). In: Proceedings of the 2nd European Semantic Web Conference (ESWC-05), number 3532 in Lecture Notes in Computer Science, pages 167–181. Springer-Verlag.
- [98] H. Stuckenschmidt and U. Visser. 2000. Semantic Translation Based on Approximate Re-Classification. In: Proceedings of the 'Semantic Approximation, Granularity and Vagueness' Workshop.
- [99] Heiner Stuckenschmidt and Frank van Harmelen. 2004. Generating and Managing Metadata for Web-Based Information Systems. Knowledge-Based Systems 17, no. 5–6, pages 201–206.
- [100] Osma Suominen. 2008. Käyttäjäkeskeinen moninäkömähaku semanttisessa portaalissa. Master of Science Thesis. Department of Computer Science, University of Helsinki.
- [101] Osma Suominen, Eero Hyvönen, Kim Viljanen, and Eija Hukka. 2009. HealthFinland—a National Semantic Publishing Network and Portal for Health Information. Journal of Web Semantics 7, no. 4, pages 271–376.
- [102] Osma Suominen, Kim Viljanen, Eero Hyvönen, Markus Holi, and Petri Lindgren. 2007. TerveSuomi.fi:n metatietomäärittely (TerveSuomi.fi metadata specification). Technical report, Semantic Computing Research Group, FinnONTO project.
- [103] M. Treglown, A. S. Pollit, M. P. Smith, P. A. J. Braekevelt, and J. E. Finlay. 1997. HIBROWSE for Bibliographic Databases: A Study of the Application of Usability Techniques in View-based Searching. Technical report, University of Huddersfield. British Library Research and Innovation Report 52.
- [104] H. Turtle and W. B. Croft. 1990. Inference Networks for Document Retrieval. In: SIGIR '90: Proceedings of the 13th Annual International ACM SIGIR

- Conference on Research and Development in Information Retrieval, pages 1–24. ACM, New York, NY, USA. ISBN 0-89791-408-2.
- [105] Octavian Udrea, V. S. Subrahmanian, and Zoran Majkic. 2006. Probabilistic RDF. In: IRI, pages 172–177.
- [106] Onni Valkeapää, Olli Alm, and Eero Hyvönen. 2007. Efficient Content Creation on the Semantic Web Using Metadata Schemas with Domain Ontology Services (System Description). In: Proceedings of the European Semantic Web Conference ESWC 2007, Innsbruck, Austria. Springer.
- [107] Antti Vehviläinen, Eero Hyvönen, and Olli Alm. 2008. A Semi-automatic Semantic Annotation and Authoring Tool for a Library Help Desk Service. IGI Group, Hershey, USA.
- [108] Kim Viljanen, Teppo Käsälä, Eero Hyvönen, and Eetu Mäkelä. 2006. ONTODELLA - A Projection and Linking Service for Semantic Web Applications. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland, pages 370–376. IEEE.
- [109] N. Walsh. 2003. RDF Twig: Accessing RDF Graphs in XSLT. In: Proceedings of Extreme Markup Languages Conference.
- [110] D. H. Widyantoro and J. Yen. 2001. A Fuzzy Ontology-based Abstract Search Engine and Its User Studies. In: The Proceedings of the 10th IEEE International Conference on Fuzzy Systems, pages 1291–1294. IEEE. ISBN 0-7803-7293-X.
- [111] J. Williamson. 2005. Bayesian Nets and Causality: Philosophical and Computational Foundations. Oxford University Press.
- [112] Sheng-Yuan Yang. 2007. An Ontology-Supported and Fully-Automatic Annotation Technology for Semantic Portals. Springer Berlin / Heidelberg.

- [113] Lofti Zadeh. 1965. Fuzzy Sets. *Information and Control* 8, pages 338–353.
- [114] Lei Zhang, Yong Yu, Jian Zhou, ChenXi Lin, and Yin Yang. 2005. An Enhanced Model for Searching in Semantic Portals. In: *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pages 453–462. ACM, New York, NY, USA. ISBN 1-59593-046-9.
- [115] H.-J. Zimmermann. 2001. *Fuzzy Set Theory and its Applications*. Springer-Verlag.

Appendix A PFSS Implementation

The logical view on the architecture of the *PFSS* framework can be seen Figure A.1.

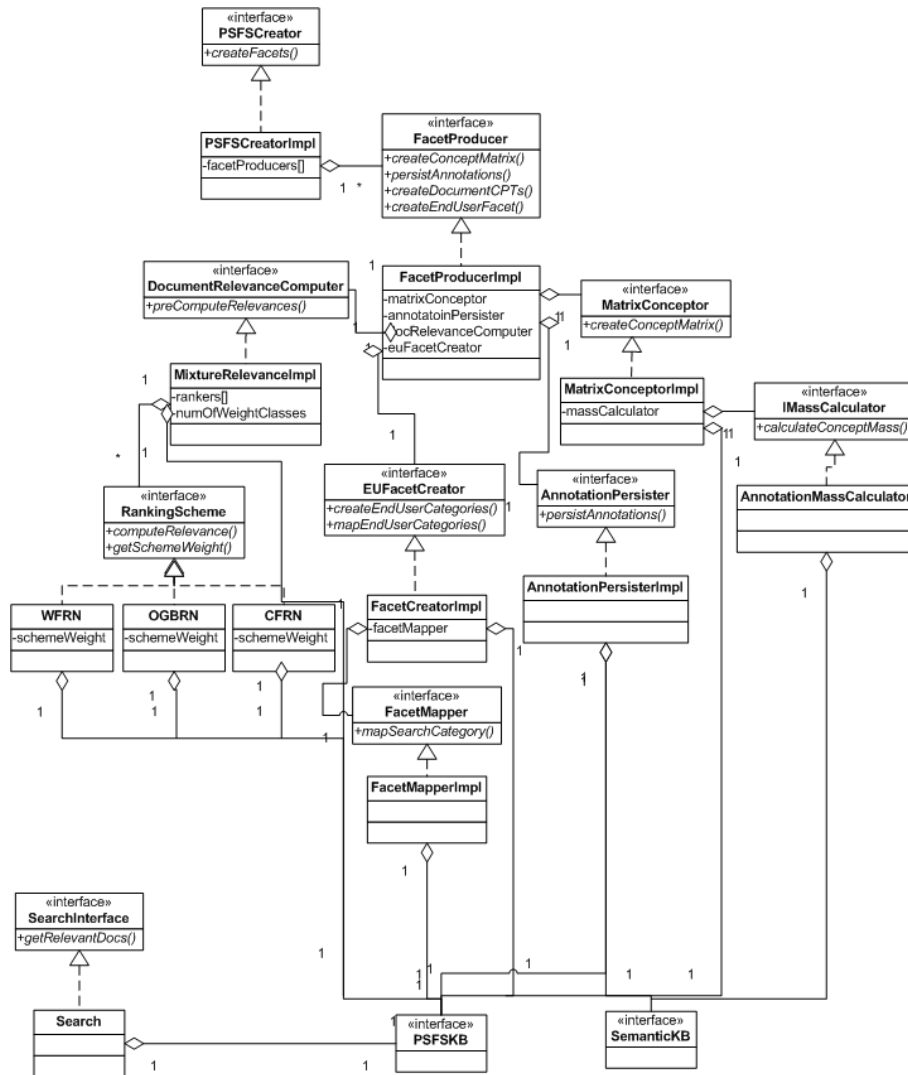


Figure A.1: The logical architecture of the *PFSS* framework

The framework is divided into two main parts. The first is the *PFSSCreator*, which is responsible for creating the *PFSS* knowledge base from the semantic knowledge base. The second part of the model is the *SearchInterface*, which returns the relevant documents ranked according to relevance for a search by a user. We will now discuss

the different modules of the *PFSS* architecture starting with the knowledge bases.

A.1 Semantic Knowledge Base

The semantic knowledge base (SKB) is implemented using RDF, and it contains at the following parts:

Annotation Ontologies The annotation ontologies can be expressed in RDFS or OWL. They contain the concept hierarchies to which the documents are annotated. The number of ontologies is not bounded, however there should be at least one ontology.

Annotation Schemas The annotation schemas are the ontologies which define how the documents are annotated. For example, they contain the definitions of the used annotation properties.

Document Annotation Instances The document annotation instances contain the actual annotations of the documents in the system. The documents are annotated according to the schemas. The annotations may be weighted.

End-user Facet Definitions End-user facet definitions are taxonomies that contain the concepts that are used as search categories. These facet definitions may have one-to-one mapping to the annotation ontologies or they may be mapped to the annotation ontologies using Boolean operators *OR*, *AND*, or *NOT*. These mappings are then handled probabilistically using the corresponding probabilistic function as described in Chapter 11. A *PFSS* Facet Definition Ontology (PFDO) has been defined to enable the smooth definition of end-user facets with the mappings to the annotation ontologies with an ontology editor such as Protege³⁴.

³⁴<http://protege.stanford.edu/>

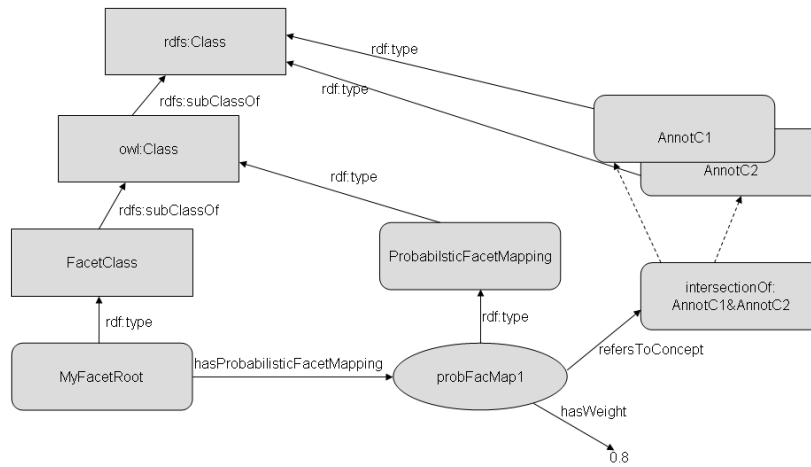


Figure A.2: PFDO with simple instance examples

PFDO defines a metaclass *FacetClass* which is a subclass of *owl : Class*. The facets are instances of *FacetClass*. In addition the ontology defines the property *hasProbabilisticFacetMapping* which is in the domain of *FacetClass*. The range of the property is the class *ProbabilisticFacetMapping* which has two properties in its domain:

refersToConcept This property points to an *rdfs:Class*. This class might be a class from the annotation ontology or it might be a class defined based on the annotation ontology classes using the *owl : intersectionOf* which represents the *and* operator, *owl : unionOf* which represents the *or* operator, or *owl : complementOf* which represents the *not* operator.

hasWeight This property has the range double. This property expresses the importance of the mapping for the facet.

Figure A.2 presents a simple example graph that adheres to this ontology.

For acquiring the annotation concept masses the SKB may contain an ontology and instance data from which the concept masses can be acquired, or they may be acquired solely based on the annotation ontology structures and the annotation instances.

A.2 PFSS Knowledge Base

The *PFSS* Knowledge base is created from the semantic knowledge base and it contains the translation of the SKB into a form suitable for the probabilistic computations according to the model presented in Chapter 11. The *PFSS* Knowledge consists of the following parts:

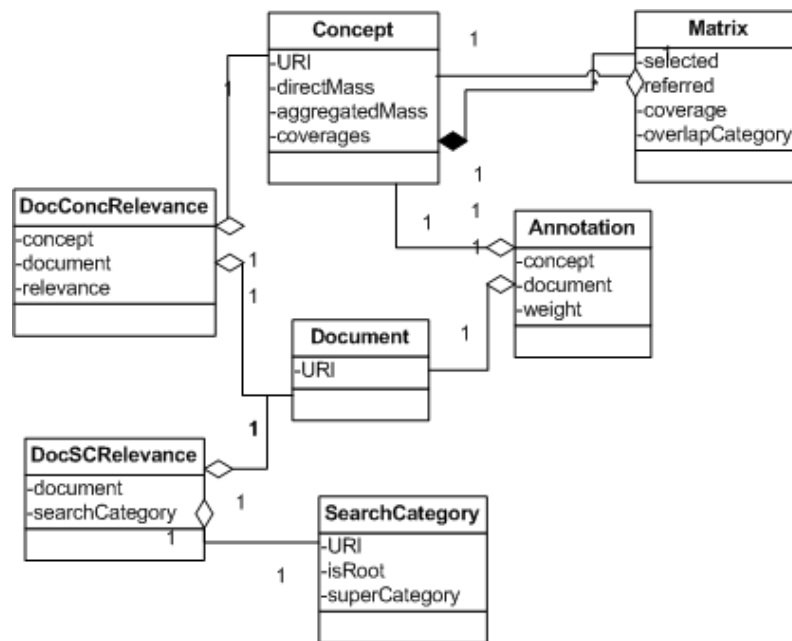


Figure A.3: The *PFSS* Knowledge base structure

Concepts The Concept representation contains the URI as well as the direct and aggregated mass of each concept. The direct mass is the mass of a concept

that is not shared with any of its subconcepts. The aggregated mass is the sum of the direct masses of all the subconcepts of the concepts in question plus the direct mass of the concept itself.

Concept overlap matrix The concept overlap matrix is a concept - concept matrix where each concept has a row and a column. Each row tells how much the concept covers the other concepts and the column tells how much the concept is covered by the other concepts. Each row also contains the information whether the covered concept is subconcept, superconcept, or partially overlapped concept.

Documents Each document representation contains simply the URI of the document.

Annotations The knowledge base contains a representation of each annotation in the system. Each annotation is represented by the annotated document, the annotation concept and the annotated weight.

Precomputed document relevances for each annotation concept These precomputed document relevances contain the relevance of each document to each annotation concept within a facet, computed according to the recursive Bayesian network model presented in Chapter 11.

Search Categories Each search category representation contains the uri, the information whether that search category is root, and references to parent and child search categories in the facet hierarchy.

Precomputed document relevances for each search category These precomputed document relevances contain the probability of relevance of each document to each search category. This relevance is computed based on the annotation concept relevances and the facet mappings according to the probabilistic model presented in Chapter 11.

Figure A.3 presents the structure of the *PFSS* knowledge base. As can be seen the only parts of the probability model presented in Chapter 11 that are not precomputed here are the multifacet-probabilities. These are easily computed on-line based on the *PFSS* knowledge base, and obviously the amount of combinations is so big that the precomputation would be unfeasible.

A.3 PFSS Creator

PFSS creator is the main interface for creating the *PFSS* KB. It has one method: *createFacets* which creates the *PFSS* KB.

A.4 Facet Producer

Facet Producer is responsible for creating a single facet into the *PFSS* KB. It contains four methods:

createConceptMatrix Creates the concept representations and concept matrix into the *PFSS* KB for this facet.

persistAnnotations Creates the document and document annotation representations into the *PFSS* KB for this facet.

createConceptRelevances Creates the representations of document relevance regarding each annotation concept in this facet.

createEUSearchCategories Creates the end-user search category representations with the relevances of each document regarding each search concept.

A.5 Matrix Conceptor

Matrix Conceptor is responsible for creating the concept representations and the concept matrix. The creation of these representations is very simple for overlap graphs. The overlap graph is transformed to a Bayesian network as described in Chapter 10 and selecting one after the other the node that represent each concept in the taxonomy and writing down in the table the posterior probabilities of the Bayesian network. The coverage matrix already is in the right representation so it will just have to be parsed.

A.6 Mass Calculator

The mass calculator is responsible for computing the direct and aggregated masses of each concept, and it also saves the computed masses in the *PFSS* KB. This can be done based on some ontological property or based on annotations as described in Chapter 10. The mass calculator offers one method: *calculateConceptMass*, which takes as an input the concept and a rule for finding the masses. This rule might be the property which points to the direct mass of each concept or a sparql rule. Basically each user of the system can implement their own masses calculator, and massify the concepts according to their wishes.

A.7 Annotation Persister

Annotation persister is responsible for creating the document and document annotation representations in *PFSS* KB. Annotation persister has one method: *persistAnnotations*. It goes through the concepts in *PFSS* KB and finds the annotations for each concept using an annotation rule that can be a single property or a

sparql rule and based on that creates document representations and the annotation representations.

A.8 Document Relevance Computer

The document relevance computer pre-computes the probabilities of relevance for documents in relation to annotation concepts and saves these relevances in *PFSS* KB. Each document relevance computer contains a list of ranking schemes which are used to achieve this. Document relevance computer offers one method: *preComputeDocumentRelevances*, and it computes the relevance of each document to each annotation concept according to the algorithms presented in Chapter 11.

A.9 Ranking Scheme

A ranking scheme implements the computations required to compute the relevance of a document according to this scheme. The ranking scheme where discussed in Chapter 11. A ranking scheme offers to methods: *computeRelevance* and *getSchemeWeight*. The first method takes as input the conceptual entity and the document under inspection and computes the relevance and the second method is used to get the weight of this scheme.

A.10 End-user Facet Creator

End-user facet creator is responsible for creating the search category representations into the *PFSS* KB. It also uses FacetMapper to precompute the probabilities of document relevance in relation to each end-user search category. It exposes

two methods: *createEndUserCategories*, and *mapEndUserCategories*. The first method goes through the facet definitions in the SKB, and creates the search category representations. The second method goes through the search categories and precomputes the probabilities of relevance of documents in relation to each search category. It uses the FacetMapper to achieve this.

A.11 Facet Mapper

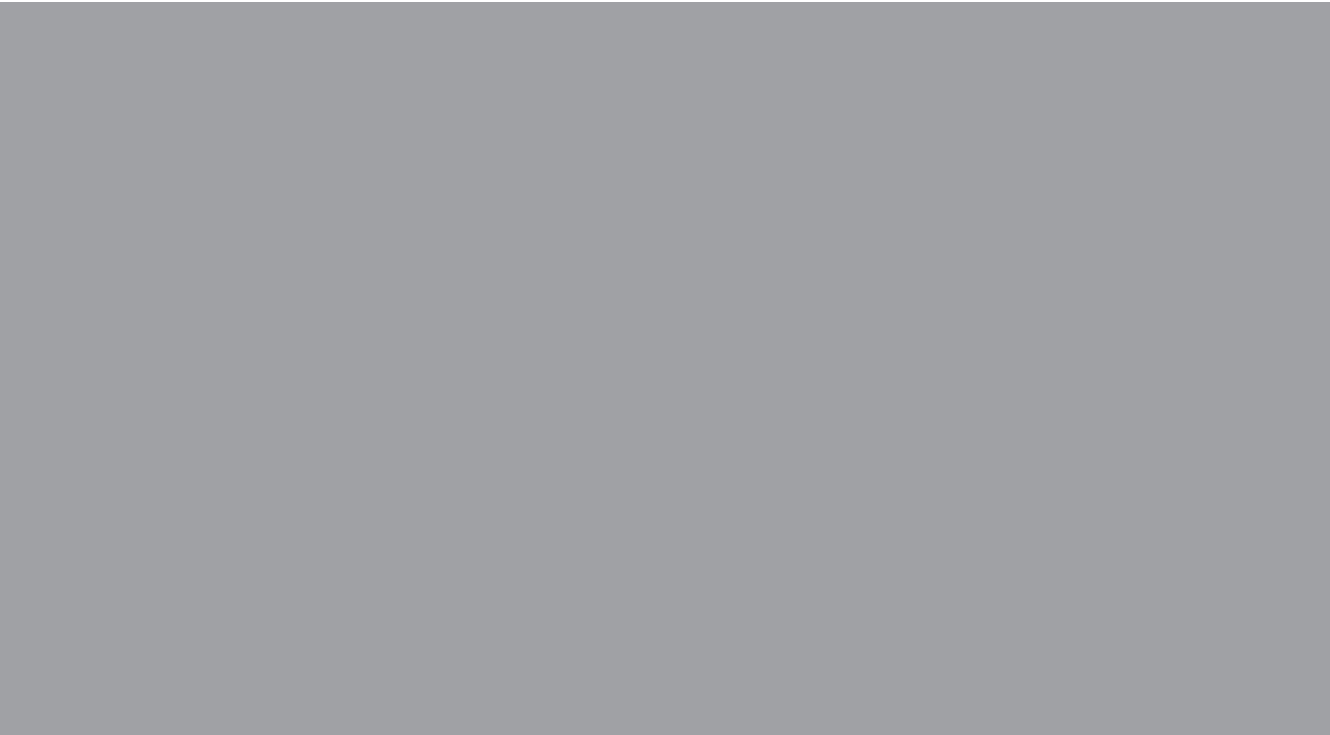
The facet mapper is responsible for precomputing the probabilities of document relevance in relation to each end-user category. This is done based on the facet definitions, i.e. mappings of search categories to annotation concepts in the SKB.

A.12 Query Interface

The query interface is responsible for returning documents to a selection of search categories from the user. The module offers one method: *getRelevantDocs*. The module makes a simple database query which returns the relevant documents. After this the system computes the combined relevances which is a product of relevance to each component search and orders the documents according to relevance.

A.13 The PFSS Container

The *PFSS* system has been implemented on Spring framework, because the IoC container offers smooth configurability and makes it very easy to plug in new components. By smooth configurability it is meant that the required input data, such as the SKB, annotation, massifying rules can be easily specified and changed.



ISBN 978-952-60-3183-5
ISBN 978-952-60-3184-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)