Integrating tf-idf Weighting with Fuzzy View-Based Search

Markus Holi and **Eero Hyvönen** and **Petri Lindgren**[1]

Abstract. This paper presents a weighting method of document annotations for fuzzy view-based semantic search (FVBSS). FVBSS is a fuzzy generalization of semantic view-based search, which supports the ranking of search results according to relevance. The presented method is an extension of the *tf-idf* weighting method. Our approach takes into account the semantic relations between indexing terms and concepts which leads to more accurate and compressed represenatation of the document and flexible information retrieval. Our preliminary user evaluation indicates that the weighting method presented here produces document rankings that match human judgment in a promissing way.

1 INTRODUCTION

Semantic portals[2] [11] usually provide the user with two basic services: 1) A search engine based on the semantics of the content [5], and 2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [6]. In this paper we concentrate on the first service, the semantic search engine.

One of the basic capabilities that are expected from a search engine is the ability to rank query results according to relevance. However, many otherwise competent semantic search engines — such as engines based on the view-based search paradigm [16, 7, 10] — do not provide this function. This follows from the fact that ontologies are based on crisp logic whereas ranking of results requires methods of uncertain reasoning.

To overcome this shortcoming we have created a fuzzy (FVBSS) generalization of the semantic view-based search paradigm, which is based on weighted document annotations [8]. FVBSS enables the ranking of search results according to relevance. This paper develops the paradigm further by presenting an automatic method to create fuzzy annotations from crisp ones. The fuzzy value reflects the relevance of the annotation to the document. The method is an ontological extension of the tf-idf [18] weighting method that is widely used in information retrieval systems.

The rest of the paper is organized as follows: In section 2 the semantic-view based search and its fuzzy extension will be described. In section 3 the ontological extension of the tf-idf weighting method will be presented. A test implementation and evaluation will be presented in section 4 and finally, section 5 summarizes the paper's contributions, discusses related work and presents learned lessons and directions for future research.

[1] Helsinki University of Technology (TKK), Media Technology and University of Helsinki, Finland, http://www.seco.tkk.fi/, email: firstname.lastname@tkk.fi

[2] See, e.g., http://www.ontoweb.org/ or http://www.semanticweb.org

2 VIEW-BASED SEARCH

2.1 Crisp View-Based Semantic Search

The view-based search paradigm[3] is based on *facet analysis* [13], 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, too, as a basis for search. The idea of the scheme is to analyze and index search items along multiple orthogonal taxonomies that are called subject *facets* or *views*. Subject headings can then be synthesized based on the analysis. This is more flexible than the traditional library classification approach of using a monolithic subject heading taxonomy.

In view-based search [16, 7, 10], the views are exposed to the end-user in order to provide her with the right query vocabulary, and for presenting the repository contents and search results along different views. The query is formulated by constraining the result set in the following way: When the user selects a category c_1 in a view v_1, the system constrains the search by leaving in the result set only such objects that are annotated (indexed) in view v_1 with c_1 or some subcategory of it. When an additional selection for a category c_2 from another view v_2 is made, the result is the intersection of the items in the selected categories, i.e., $c_1 \cap c_2$. After the result set is calculated, it can be presented to the end-user according to the view hierarchies for better readability. This is in contrast with traditional search where results are typically presented as a list of decreasing relevance.

View-based search has been integrated with the notion of ontologies and the semantic web [10, 15, 9, 12]. The idea of such *semantic view-based search* is 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 can 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 result set.

2.2 Fuzzy Semantic View-Based Search

The view-based search scheme has also some shortcomings. First, it does not incorporate the notion of relevance. Thus, there is not a way to rank the results according to relevance. Second, in semantic view-based search the views are generated according to the concept

[3] A short history of the parading is presented in http://www.view-based-systems.com/history.asp

hierarchies of the ontologies. This is not always ideal from the viewpoint of the end user, because the ontologies usually are created by and for domain experts, and thus it contains concepts and structures that might not be familiar or intuitive to a non-professional end-user.

To overcome these problems we created a fuzzy version of the search paradigm [8]. In this paradigm 1) the degrees of relevance of documents can be determined and 2) distinct end-user's views to search items can be created and mapped onto indexing ontologies and the underlying search items (documents). The framework generalizes view-based search from using crisp sets to fuzzy set theory and is called *fuzzy view-based semantic search.*

2.2.1 The Architecture of the Framework

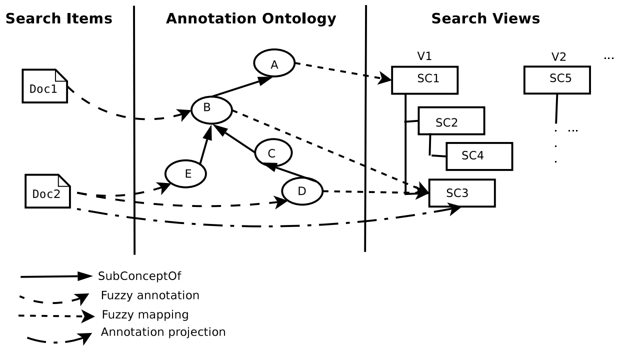

Figure 1. Components of the fuzzy view-based semantic framework

The architecture of the framework is depicted in figure 1. The framework consists of the following components:

Search Items The search items are a finite set of documents D depicted on the left. D is the fundamental set of the fuzzy view-based search framework.

Annotation Ontology The search items are annotated according to the ontology by the indexer. The ontology consists of two parts. First, a finite set of annotation concepts AC, i.e. a set of fuzzy subsets of D. Annotation concepts $AC_i \in AC$ are atomic. Second, a finite set of annotation concept inclusion axioms $AC_i \subseteq AC_j$[4], where $AC_i, AC_j \in AC$ are annotation concepts and $i, j \in N$, and $i \neq j$. These inclusion axioms denote subsumption between the concepts and they constitute a concept hierarchy.

Search Views Search views are hierarchically organized search categories for the end-user to use during searching. The views are created and organized with end-user interaction in mind, and may not be identical to the annotation concepts for professional indexers. Each search category SC_i is a fuzzy subset of D. In crisp view-based search the intersection of documents related to selected search categories is returned as the result set, while in fuzzy view-based search, 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 result R is not a crisp set of search items $R = SC_1 \cap ... \cap SC_n = \{Doc_1, ..., Doc_m\}$ as in view-based search, but a fuzzy set where the relevance of each item is specified by the value of the membership function mapping:

$$R = SC_1 \cap ... \cap SC_n = \{(Doc_1, \mu_1), ..., (Doc, \mu_m)\}$$

In the following the required mappings are described. For a fuller description see [8].

2.2.2 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. In (semantic) view-based search, the annotation of a search item is the crisp set of annotation concept categories in which the item belongs to. In figure 1, annotations are represented using bending dashed arcs from *Search Items* to *Annotation Ontology*. For example, the annotation of the item *Doc2* would be the set $A_{Doc2} = \{E, D\}$.

In our approach, the relevance of different annotation concepts with respect to a document may vary and is represented by a *fuzzy annotation.* The fuzzy annotation A_D of a document D is the set of its fuzzy concept membership assertions:

$$A_D = \{(AC_1, \mu_1), ..., (AC_n, \mu_n)\} \text{ where } \mu_i \in (0, 1]$$

Here μ_i tell the degrees by which the annotated document is related to annotation concepts AC_i. For example, our test dataset consisted of health related documents that were annotated using the finnish translation of Medical Subject Headings (MeSH)[5]. A document D_1 from that document set was given the fuzzy annotation

$$A_{D_1} = \{(Exercise, 0.3), (Diet, 0.4)\}$$

Based on the 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 [17]. Thus, applying this principle to fuzzy sets 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.

$$\forall D_i \in D, \mu_{AC_j}(D_i) = max(\mu_{AC_i}(D_i))$$

where $AC_i \subseteq AC_j$.

For example, assume that we have a document D_2 that is annotated with the annotation concept *Asthma* with weight 0.8, i.e. $\mu_{Asthma}(D_2) = 0.8$. Assume further, that in the annotation ontology *Asthma* is a subconcept of *Diseases*, i.e. $Asthma \subseteq Diseases$. Then,

$$\mu_{Diseases}(D_2) = \mu_{Asthma}(D_2) = 0.8$$

2.2.3 Fuzzy Mappings

Each search category SC_i in a view V_j is defined using concepts from the annotation ontology by a finite set of fuzzy concept inclusion axioms that we call *fuzzy mappings*:

---

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

5 http://http://www.nlm.nih.gov/mesh/

$$AC_i \subseteq_\mu SC_j \quad \text{where } AC_i \in AC, SC_j \in V_k,\ i,j,k \in N$$
$$\text{and } \mu \in (0,1]$$

A fuzzy mapping describes the meaning of a search category SC_j by telling to what degree μ the membership of a document D_i in an annotation concept AC_i implies its membership in SC_j. Intuitively, a fuzzy mapping reveals to which degree the annotation concept can be considered a subconcept of the search category. In figure 1, fuzzy mappings are represented using straight dashed arcs.

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 [20, 4]. We use Goguen's fuzzy implication, i.e.

$$i(\mu_{AC_j}(D_i), \mu_{SC_i}(D_i)) = 1 \text{ if } \mu_{SC_i}(D_i) \geq \mu_{AC_j}(D_i)$$
$$\text{else } \mu_{SC_i}(D_i)/\mu_{AC_j}(D_i) \qquad \forall D_i \in D$$

Let us continue with the example case in the end of section 2.2.2 where we defined the membership of document D_1 in the annotation concept $Diseases$. Assume that we want to define a search category *Food and Diseases* that will give the user information about the relation of diseases to food.

As part of the category definition we would create a fuzzy mapping

$$Diseases \subseteq_{0.1} \textit{Food and Diseases}$$

Based on this fuzzy mapping, the membership degree of the document $D1$ in *Food and Diseases* is

$$\mu_{\textit{Food and Diseases}}(D1) = \mu_{Diseases}(D1)*0.1 = 0.8*0.1 = 0.08$$

A search category SC_j is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Fuzzy mappings can be created by a human expert or by an automatic or a semi-automatic ontology mapping tool.

Mappings can be nested. Two fuzzy mappings $M_1 = AC_i \subseteq_\mu SC_i$ and $M_2 = AC_j \subseteq_\nu SC_i$ are *nested* if $AC_i \subseteq AC_j$, i.e., if they point to the same search category, and one of the involved annotation concepts is the subconcept of the other. In this case the more specific mapping is relevant for a document when computing its membership in the search category SC_j.

It is also possible to map a search category to a Boolean combination of annotation concepts. In this cases the membership functions of these boolean concepts are calculated according to the fuzzy union, intersection and negation operations.

2.2.4 Performing the Search

In view-based search the user can query by choosing concepts from the views. In crisp semantic view-based search, the extension E of a search category is the union of its projection P and the extensions of its subcategories S_i, i.e. $E = P \bigcup S_i$. The result set R to the query is simply the intersection of the extensions of the selected search categories $R = \bigcap E_i$ [9].

In fuzzy view-based search we extend the crisp union and intersection operations to fuzzy intersection and fuzzy union. Recall, from section 2.2.3 that a search category was defined as the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Thus, the fuzzy union part of the view-based search is already taken care of. Now, if E is the set of selected search categories, then the fuzzy result set R is the fuzzy intersection of the members of E, i.e. $R = SC_1 \cap ... \cap SC_n$, where $SC_i \in E$.

Using Gödel's intersection [24], we have:

$$\mu_R(D_k) = min(\mu_{SC_1}(D_k), ..., \mu_{SC_n}(D_k)) \forall D_k \in D$$

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

3 AUTOMATIC CREATION OF FUZZY ANNOTATIONS

We created the fuzzy annotations with an ontological extension of the tf-idf weighting method. In the following first the tf-idf weighting method is described and then our ontological extension of it is presented.

3.1 Tf-idf

The tf-idf [18] (term frequency - inverse document fequency) 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 weight[6] 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.

$$idf_{t_i} = log(\frac{N}{df_{t_i}})$$

The normalized tf-idf weight of the term t_i in document Doc_j is

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

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.

3.2 Ontological Extension of tf-idf

We extended the tf-idf weighting method so that it can be used to weight existing crisp document annotations. Recall from chapter 2.2.2 that a crisp annotation of a document Doc_j is the set $A_{Doc_j} = \{AC_1, ..., AC_n\}$, where $AC_1, ..., AC_n$ are concepts of

[6] See, http://www.sims.berkeley.edu:8000/courses/is202/f05/LectureNotes/202-20051110.pdf

the annotation ontology. The weighting is done based on the textual content of the document and the description of each concept $AC_1, ..., AC_n$ 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

$$idf_{AC_i} = log(\frac{N}{df_{AC_i}})$$

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

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

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.

4 TEST IMPLEMENTATION AND EVALUATION

We implemented the representation of ontologies, annotations and search views using RDF [1]. The algorithms were implemented using Java[7] and its Semantic Web Framework Jena[8]. Next we will describe the document collection, the ontology and the search views of our test implementation and then a preliminary evaluation of the method will be presented.

4.1 Document Collection and Ontology

Our document set consisted of 163 documents from the web site of the National Public Health Institute[9] of Finland (NPHI).

As an annotation ontology we created a SKOS [2] version of FinMeSH, the Finnish translation of MeSH. 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 extension of tf-idf described above. The search views with the mappings were designed and created by hand.

4.2 Evaluation

The main practical contribution of our framework in comparison to crisp view-based search is the ranking of search results according to relevance. A preliminary user-test was conducted to evaluate the ranking done by the implementation described above. The test group consisted of five subjects.

The test data was created in the following way. 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 who's 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 users were 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.

5 DISCUSSION

This paper presented an ontological extension to the widely used tf-idf weighting method. It was designed to enable the automatic weighting of crisp document annotations based on the textual content of each document and the conceptual information of the ontology. The ontological tf-idf method evaluates the importance of the annotation concept to the document.

5.1 Contributions

The main benefits of the method when compared to the traditional tf-idf method are: 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.

We integrated the method to our FVBSS framework as a way to automatically create fuzzy annotations from crisp annotations. FVBSS enables the ranking of search results according to query relevance in view-based semantic search. A prototype implementation and its application to a data set in semantic eHealth portal was discussed and evaluated.

5.2 Related Work

The fuzzy semantic view-based search framework presented in this paper generalizes the traditional view-based search paradigm [16, 7,

[7] http://java.sun.com
[8] http://jena.sourceforge.net
[9] http://www.ktl.fi/

10] and its semantic extension developed in [10, 15, 9, 12]. Fuzzy reasoning [22] has been used before in IR but to our knowledge not with (semantic) view-based search.

We have applied the idea presented by Straccia [20] in his fuzzy extension to the description logic *SHOIN(D)* and Bordogna [4] of using fuzzy implication to model fuzzy inclusion between fuzzy sets. Also other fuzzy extensions to description logic exist, such as [19, 14].

Zhang et al. [23] 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. 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 by providing an end-user specific view to the search items.

Akrivas et al. [3] present an interesting method for context sensitive semantic query expansion. In this method, 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 [3], 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. Thus, the difference to our work is that the inclusion relation in in the fuzzy hierarchy of [3] is crisp, whereas in our *fuzzy mappings* the inclusion relation itself is fuzzy.

Widyantoro and Yen [21] 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. The fuzzy ontology of Widyantoro and Yen is based on a set of documents, and works on that document set. The automatic creation of ontologies is an interesting issue by itself, but it is not considered in our paper. At the moment, better and richer ontologies can be built by domain specialists than by automated methods.

5.3 Lessons Learned and Future Work

The ontological extension of the tf-idf weighting method proved to be rather straight forward to design and implement. Our preliminary evaluation of ranking search results with the framework were promising. However, the number of test subjects and the size of test data set was still too small for proper statistical analysis.

Our framework did get some inspiration from 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 [20] 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.

Furthermore, the datasets in our *Tervesuomi.fi* eHealth portal case study are large. The number of search-items will be probably between 50,000 and 100,000, and the number of annotation concepts probably between 40,000 and 50,000. For this reason we wanted to build our framework on the view-based search paradigm that has proven to be scalable to relatively large data sets. For example, the semantic view-based search engine *OntoViews* was tested to scale up to 2.3 million search items and 275,000 search categories in [12]. The fuzzy generalization adds only a constant coefficient to the computational complexity of the paradigm.

In the future we intend to implement the framework with a larger dataset in the semantic *Tervesuomi.fi* eHealth portal and test it with a larger user group. The fuzzy framework will be attached to the *OntoViews* tool as a separate ranking module. Thus, there is not a need for major refactoring of the search engine in *OntoViews*.

In addition we intend to apply the framework to the ranking of the recommendation links created by *OntoDella*, which is the semantic recommendation service module of *OntoViews*.

ACKNOWLEDGEMENTS

Our research was funded mainly by the National Technology Agency Tekes. The National Public Health Institute of Finland (NPHI) provided us with the data annotated by Johanna Eerola.

REFERENCES

[1] *RDF Primer*. http://www.w3.org/TR/rdf-primer.

[2] *SKOS Core Guide*, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.

[3] G. Akrivas, M. Wallace, G. Andreou, G. Stamou, and S. Kollias, 'Context - sensitive semantic query expansion', in *Proceedings of the IEEE International Conferrence on Artificial Intelligence Systems (ICAIS)*, (2002).

[4] G. Bordogna, P. Bosc, and G. Pasi, 'Fuzzy inclusion in database and information retrieval query interpretation', in *ACM Computing Week - SAC'96*, Philadelphia, USA, (1996).

[5] S. Decker, M. Erdmann, D. Fensel, and R. Studer, 'Ontobroker: Ontology based access to distributed and semi-stuctured information', *DS-8*, 351–369, (1999). http://citeseer.nj.nec.com/article/decker98ontobroker.html.

[6] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall, 'Conceptual open hypermedia = the semantic web', in *Proceedings of the WWW2001, Semantic Web Workshop*, Hongkong, (2001).

[7] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee, 'Finding the flow in web site search', *CACM*, **45**(9), 42–49, (2002).

[8] M. Holi and E. Hyvönen, 'Fuzzy view-based semantic search', *Paper, submitted*, (2006). http://www.seco.tkk.fi/u/mholi/papers/fuzzyViewSearch.pdf.

[9] Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula, 'Museumfinland – finnish museums on the semantic web', *Journal of Web Semantics*, **3**(2), 25, (2005).

[10] Eero Hyvönen, Samppa Saarela, and Kim Viljanen, '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)*, (2004).

[11] A. Maedche, S. Staab, N. Stojanovic, R. Struder, and Y. Sure, 'Semantic portal - the seal approach', Technical report, Institute AIFB, University of Karlsruhe, Germany, (2001).

[12] Eetu Makelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen, 'Ontoviews – a tool for creating semantic web portals', in *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, (May 2004).

[13] A. Maple. Faceted access: a review of the literature, 1995. http://library.music.indiana.edu/tech_s/mla/facacc.rev.

[14] M. Mazzieri and A. F. Dragoni, 'Fuzzy semantics for semantic web languages', in *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, (Nov 2005).

[15] Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff, 'View-based user interfaces for information retrieval on the semantic web', in *Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction*, (Nov 2005).

[16] A. S. Pollitt, 'The key role of classification and indexing in view-based searching', Technical report, University of Huddersfield, UK, (1998). http://www.ifla.org/IV/ifla63/63polst.pdf.

[17] A. Rector, 'Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases', in *Proceedings of Pacific Symposium on Biocomputing*, (2004).

[18] G. Salton and C. Buckley, 'Term weighting approaches in automatic text retrieval', Technical report, Ithaca, NY, USA, (1987).

[19] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks, 'The fuzzy description logic f-shin', in *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, (Nov 2005).

[20] Umberto Straccia, 'Towards a fuzzy description logic for the semantic web (preliminary report)', in *2nd European Semantic Web Conference (ESWC-05)*, number 3532 in Lecture Notes in Computer Science, pp. 167–181, Crete, (2005). Springer Verlag.

[21] D.H. Widyantoro and J. Yen, 'A fuzzy ontology-based abstract seawrch engine and its user studies', in *The Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, (2002).

[22] L. Zadeh, 'Fuzzy sets', *Information and Control*, (1965).

[23] L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang, 'An enhanced model for searching in semantic portals', in *Proceedings of the Fourteenth International World Wide Web Conference*, (May 2005).

[24] H.-J. Zimmermann, *Fuzzy Set Theory and its Applications*, Springer, 2001.