Fuzzy View-Based Semantic Search

Markus Holi and Eero Hyvönen

Helsinki University of Technology (TKK), Media Technology and University of Helsinki
P.O. Box 5500, FI-02015 TKK, FINLAND,
http://www.seco.tkk.fi/
email: firstname.lastname@tkk.fi

Abstract. This paper presents a fuzzy version of the semantic view-based search paradigm. Our framework contributes to previous work in two ways: First, the fuzzification introduces the notion of relevance to view-based search by enabling the ranking of search results. Second, the framework makes it possible to separate the end-user's views from content indexer's taxonomies or ontologies. In this way, search queries can be formulated and results organized using intuitive categories that are different from the semantically complicated indexing concepts. The fuzziness is the result of allowing more accurate weighted annotations and fuzzy mappings between search categories and annotation ontologies. A prototype implementation of the framework is presented and its application to a data set in a semantic eHealth portal discussed.

1 Introduction

Much of semantic web content will be published using semantic portals[1] [16]. Such portals usually provide the user with two basic services: 1) A search engine based on the semantics of the content [6], and 2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [9]. In this paper we concentrate on the first service, the semantic search engine.

1.1 View-Based Semantic Search

The view-based search paradigm[2] [23, 11, 13] is based on *facet analysis* [18], 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, 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

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

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

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 sub-category 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 [13, 21, 12, 17]. 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 could be defined using logic rules. This facilitated more intelligent "semantic" search of indirectly related items. Another benefit is that the logic layer of rules made 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.

1.2 Problems of View-Based Search

View-based search helps the user in formulating the query in a natural way, and in presenting the results along the views. The scheme has also some shortcomings. In this paper we consider two of them:

Representing relevance View-based search does not incorporate the notion of relevance. In view-based search, search items are either annotated using the categories or mapped on 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 sub-concepts. There is no way to rank the results according to their relevance as in traditional search. For example, consider two health-related documents annotated with the category Helsinki. One of the documents could describe the health services in Helsinki, the other could be a European study about alcohol withdrawal syndromes of heavy alcohol users, for which the research subject were taken randomly from London, Paris, Berlin, Warsaw and Helsinki. It is likely that the first document is much more relevant for a person interested in health and Helsinki.

Separating end-user's views from indexing schemes Annotation concepts used in annotation taxonomies or ontologies often consist of complicated professional concepts needed for accurate indexing. When using ontologies, the annotation concepts are often organized according a formal division of the topics or based on an upper-ontology. This is important because it enables automatic reasoning over the ontologies. However, such categorizations are not necessarily useful as search views because they can be difficult to understand and too detailed from the human end-users viewpoint. The user then needs a view to the content that is different from the machine's or indexer's viewpoint. However, current view-based system do not differentiate between indexer's, machine's, and end-user's views. In our case study,

for example, we deal with problem of publishing health content to ordinary citizens in a coming semantic portal *Tervesuomi.fi*. Much of the material to be used has been indexed using complicated medical terms and classifications, such as Medical Subject Headings[3] (MeSH). Since the end-user is not an expert of the domain and is not familiar with the professional terms used in the ontology, their hierarchical organization is not suitable for formulating end-user queries or presenting the result set, but only for indexing and machine processing.

This paper presents a fuzzy version of the semantic view-based search paradigm in which 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*. In the following, this scheme is first developed using examples from the *Tervesuomi.fi* portal content. After this an implementation of the system is presented. In conclusion, contributions of the work are summarized, related work discussed, and directions for further research proposed.

2 Fuzzy View-based Semantic Search

2.1 Architecture of the Framework

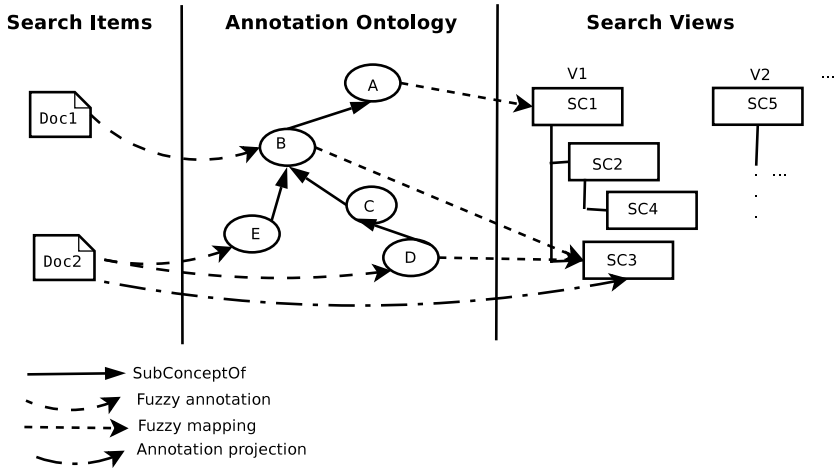


Fig. 1. Components of fuzzy view-based semantic search framework.

Figure 1 depicts the architecture of the fuzzy view-based semantic search framework. The framework consists of the following components:

[3] http://www.nlm.nih.gov/mesh/

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. 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 in detail.

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 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:

$A_D = \{(AC_1, \mu_1), ..., (AC_n, \mu_n)\}$, 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;

$A_{D1} = \{(Exercise, 0.3), (Diet, 0.4)\}$

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

[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.

implies being in the superconcept [24]. 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 $\nu \in (0, 1]$, and 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.

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

For example, assume that we have a document $D1$ that is annotated with annotation concept *Asthma* with weight 0.8, i.e. $\mu_{Asthma}(D1) = 0.8$. Assume further, that in the annotation ontology *Asthma* is a subconcept of *Diseases*, i.e. $Asthma \subseteq Diseases$. Then,

$\mu_{Diseases}(D1) = \mu_{Asthma}(D1) = 0.8$.

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*:

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

A fuzzy mapping constrains 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.

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 [27, 5]. We use Goguen's fuzzy implication, i.e.

$i(\mu_{AC_j}(D_i), \mu_{SC_i}(D_i)) = 1$ if $\mu_{SC_i}(D_i) \geq \mu_{AC_j}(D_i)$, and $\mu_{SC_i}(D_i)/\mu_{AC_j}(D_i)$ otherwise, $\forall D_i \in D$.

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

A search category SC_j is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Using Gödel's union function[5] the membership function of SC_j is

$\mu_{SC_j}(D_i) = max(\mu_{SC_1}(D_i), ..., \mu_{SC_n}(D_i), \mu_{MS_1}(D_i), ..., \mu_{MS_n}(D_i)), \forall D_i \in D$, where $SC_{1,...,n}$ are subcategories of SC_j, and $MS_{1,...,n}$ are the sets defined by the fuzzy mappings pointing to SC_j. This extends the idea of view-based search, where view categories correspond directly to annotation concepts.

Continuing with the example case in the end of section 2.2 where we defined the membership of document D_1 in the annotation concept $Diseases$. If we have a fuzzy mapping

$Diseases \subseteq_{0.1} Food\&Diseases$

then the membership degree of the document $D1$ in $Food\&Diseases$ is

$\mu_{Food\&Diseases}(D1) = \mu_{Diseases}(D1) * 0.1 = 0.8 * 0.1 = 0.08$.

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

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 1, fuzzy mappings are represented using straight dashed arcs.

The fuzzy mappings of a search category 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. Nesting between the fuzzy mappings M_1 and M_2 is interpreted as a shorthand for $M_1 = AC_i \subseteq_\mu SC_i$ and $M_2 = (AC_j \cap \neg AC_i) \subseteq_\nu SC_i$. This interpretation actually dissolves the nesting. For example, if we have mappings

$M_1 =$ *Animal nutrition* $\subseteq_{0.1} Nutrition_{sc}$ and $M_2 =$ *Nutrition* $\subseteq_{0.9} Nutrition_{sc}$, and in the annotation ontology *Animal nutrition* $\subseteq$ *Nutrition*, then M_1 is actually interpreted as

$M_1 =$ *Nutrition* $\cap \neg Animalnutrition \subseteq_{0.9} Nutrition_{sc}$.

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 search view 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 $AC_1 \cap ... \cap AC_n$ (intersection), $AC_1 \cup ... \cup AC_n$ (union) or $\neg AC_1$ (negation), where $AC_1, ..., AC_n \in AC$.

In the following, 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 2 will used as examples in the text. In section 2.5 we describe how to execute the view-based search based on the projected annotations and end-user's selections.

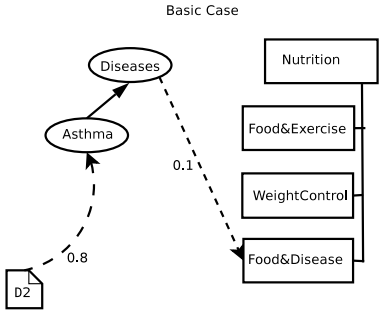
2.4 Mappings to Boolean Concepts

In the following, the membership function definition for each type of Boolean concept is listed, according to the widely used Gödel's functions[6]:

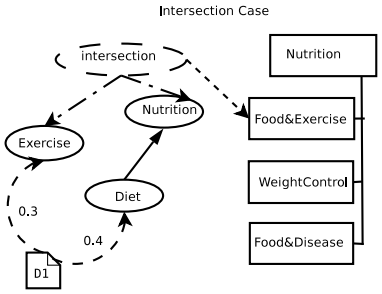
Union Case $AC_j = AC_k \cup ... \cup AC_n$: The membership degree of a document in AC_j is the maximum of its concept membership values in any of the components of the union concept:
$\forall D_k \in D, \mu_{AC_j}(D_k) = max(\mu_{AC_i}(D_k))$, where $i \in k, ..., n$
In the example union case of figure 2(c) we get
$\mu_{Thinnes \cup Obesity}(D5) = max(\mu_{Thinnes}(D5), \mu_{Obesity}(D5))$
$= max(0, 0.8) = 0.8$.

Intersection Case $AC_j = AC_k \cap ... \cap AC_n$: 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. $\forall D_k \in D, \mu_{AC_j}(D_k) = min(\mu_{AC_i}(D_k))$, where $i \in k, ..., n$.
In the example intersection case of figure 2(b) we get

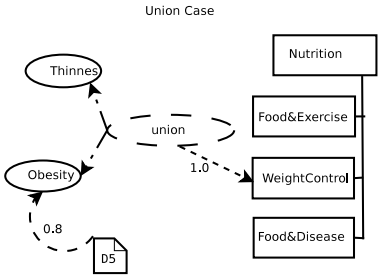
[6] If A and B are fuzzy sets of the fundamental set X, then $\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x))$, $\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x))$, and $\mu_{\neg A}(x) = 0$, if $\mu_A(x) \geq 0$, 0 otherwise, $\forall x \in X$.



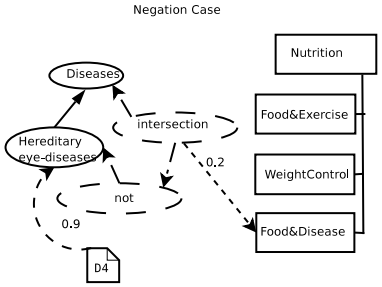
(a) Basic case: the referred concept is an atomic annotation concept.



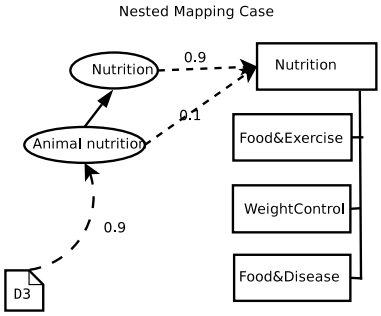
(b) Intersection case: the rereferred concept is an intersection of annotation concepts.



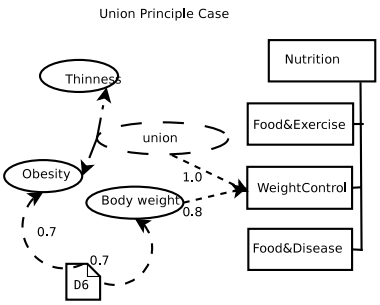
(c) Union case: the referred concept is a union of annotation concepts.



(d) Negation case: the referred concept is a complement of annotation an annotation concept.



(e) Nested mappings case: two fuzzy mappings are nested.



(f) Union Principle case: the definition of a search category is the union of its fuzzy mappings.

Fig. 2. Real-world examples of annotation projection cases

$\mu_{\textit{Nutrition}\cap\textit{Exercise}}(D1) = min(\mu_{\textit{Nutrition}}(D1), \mu_{\textit{Exercise}}(D1))$
$= min(0.4, 0.3) = 0.3$.

Negation Case $AC_j = \neg AC_k$: 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. $\forall D_k \in D, \mu_{AC_j}(D_k) = 0$ if $\mu_{AC_k}(D_k) > 0$ and $\mu_{AC_j}(D_k) = 1$ if $\mu_{AC_k}(D_k) = 0$. In the example negation case of figure 2(d) we get
$\mu_{\neg\textit{Congenital diseases}}(D4) = 0$ because $(\mu_{\textit{Congenital diseases}}(D4) = 0.9) > 0$.

After the membership function of each boolean concept is defined, the membership function of the search concept can be computed based on the fuzzy mappings. For example, in figure 2(f) the projection of document $D6$ to the search view is done in the following way: The membership degrees of $D6$ in the relevant annotation concepts are
$\mu_{\textit{Thinness}\cup\textit{Obesity}}(D6) = 0.7$ and $\mu_{\textit{Body weight}}(D6) = 0.7$.
Now, the first fuzzy mapping of these yields
$\mu_{MS_1}(D6) = 0.7$
and the second one
$\mu_{MS_2}(D6) = 0.7 * 0.8 = 0.56$.
Because each search category is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it, and $WeightControl$ does not have any subcategories, we get
$\mu_{WeightControl}(D6) = max(\mu_{MS_1}(D6), \mu_{MS_2}(D6)) = 0.7$.

2.5 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 $R = \bigcap E_i$ [12].

In fuzzy view-based search we extend the crisp union and intersection operations to fuzzy intersection and fuzzy union. Recall, from section 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 [32], 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 Implementation

In the following an implementation of our framework is presented. In sections 3.1 and 3.2, RDF [1] representations of fuzzy annotations and search views are described, respectively. Section 3.3 presents an algorithm for the annotation projection discussed in

section 2.4. Section 3.4 describes the dataset that we used to test the framework, and finally, in section 3.5 preliminary user evaluation of our test implementation is presented.

3.1 Representing Fuzzy Annotations

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 annotations 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 2 is represented in the following way.

```
<DescriptorElement rdf:ID="descriptorelement_63">
     <hasTerm rdf:resource="&mesh;D004032"/>
     <hasWeight>0.4</hasWeight>
</DescriptorElement>
<DescriptorElement rdf:ID="descriptorelement_64">
     <hasTerm rdf:resource="&mesh;D015444"/>
     <hasWeight>0.3</hasWeight>
</DescriptorElement>
<Descriptor rdf:ID="Descriptor_6">
     <describes rdf:resource="#D1"/>
     <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.

Our model does not make any commitments about the method by which these fuzzy annotations are created.

3.2 Representing Search Views

We created an RDF representation of the views and the mappings between the search categories of the views and the annotation concepts. Our representation is based on the Simple Knowledge Organization System (SKOS) [3, 2]. For example the search categories $Nutrition$ and $Nutrition\&Diseases$ in figure 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 it's 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.

Our model does not make any commitments about the method by which these fuzzy mappings are created.

3.3 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 Framework[7]. 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 dissolved. This is done by running through the mappings that point to each search category, detecting the nested mappings using the concept hierarchy and dissolving the nesting according to the method described in section 2.3.
3. The memberhsip function of each annotation concept is computed using the method described in section 2.2.
4. The membership function of each search category is computed using the method described in section 2.3.

[7] http://jena.sourceforge.net/

3.4 Dataset and Ontology

Our document set consisted of 163 documents from the web site of the National Public Health Institute[8] of Finland (NPHI).

As an annotation ontology we created a SKOS translation 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 version of the TF-IDF [25] weighting method widely used in IR systems. 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 search views with the mappings were designed and created by hand.

3.5 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.

4 Discussion

This paper presented a fuzzy generalization to the view-based semantic search paradigm. A prototype implementation and its application to a data set in semantic eHealth portal was discussed and evaluated.

[8] See, http://www.ktl.fi/

4.1 Contributions

The presented fuzzy view-based search method provides the following benefits when in comparison with the crisp view-based search:

Ranking of the result set Traditional view-based 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 straightforward based on the membership functions of the concepts.

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. 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.

No commitment to any particular implementation or weighting scheme The paper presents a generic framework to include uncertainty and vagueness in view-based search. It can be implemented in many different ways, as long as the weighting or ranking methods can be mapped to fuzzy values in the range (0,1].

4.2 Related Work

The work in this paper generalizes the traditional view-based search paradigm [23, 11, 13] and its semantic extension developed in [13, 21, 12, 17].

The problem of representing vagueness and uncertainty in ontologies has been tackled before. In methods using rough sets [28, 22] only a rough, egg-yolk representation of the concepts can be created. Fuzzy logic [30], allows for a more realistic representation of the world.

Also probabilistic methods have been developed for managing uncertainty in ontologies. Ding and Peng [7] present principles and methods to convert an OWL ontology into a Bayesian network. Their methods are based on probabilistic extensions to description logics [15, 8]. Also other approaches for combining Bayesian networks and ontologies exist. Gu [10] 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 [20] presents a probabilistic ontology mapping tool. In this approach the nodes of the Bayesian network represent matches between pairs of classes in the two ontologies to be mapped. The arrows of the BN are dependencies between matches.

Kauppinen and Hyvönen [14] present a method for modeling partial overlap between versions of a concept that changes over long periods of time.

Our method is based on fuzzy logic [30]. We have applied the idea presented by Straccia [27] in his fuzzy extension to the description logic *SHOIN(D)* and Bordogna [5] of using fuzzy implication to model fuzzy inclusion between fuzzy sets. Also other fuzzy extensions to description logic exist, such as [26, 19].

Zhang et al. [31] 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. [4] 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 [4], 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 fuzzy inclusion was interpreted as fuzzy implication, meaning that the inclusion relation itself is partial.

Widyantoro and Yen [29] 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.

4.3 Lessons Learned and Future Work

The fuzzy generalization of the (semantic) view-based search paradigm proved to be rather straight forward to design and implement. Crisp view-based search is a special case of the fuzzy framework such that the annotations and the mappings have the weight 1.0, i.e. are crisp.

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 [27] 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 [17]. 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*.

Acknowledgments 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 Mapping Vocabulary Specification*, 2004. http://www.w3.org/2004/02/skos/mapping/spec/.
3. *SKOS Core Guide*, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.
4. 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.
5. 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.
6. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-stuctured information. *DS-8*, pages 351–369, 1999. http://citeseer.nj.nec.com/article/decker98ontobroker.html.
7. Z. Ding and Y. Peng. A probabilistic extension to ontology language owl. In *Proceedings of the Hawai'i Internationa Conference on System Sciences*, 2004.
8. R. Giugno and T. Lukasiewicz. 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, 2002.
9. 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.
10. T. Gu and D.Q. Zhang H.K. Pung. A bayesian approach for dealing with uncertain contexts. In *Advances in Pervasive Computing*, 2004.
11. 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.
12. 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.
13. 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.

14. T. Kauppinen and E. Hyvönen. Geo-spatial reasoning over ontology changes in time. In *Proceedings of IJCAI-2005 Workshop on Spatial and Temporal Reasoning*, 2005.
15. D. Koller, A. Levy, and A. Pfeffer. P-classic: A tractable probabilistic description logic. In *Proceedings of AAAI-97*, 1997.
16. 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.
17. 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.
18. A. Maple. Faceted access: a review of the literature, 1995. http://library.music.indiana.edu/tech_s/mla/facacc.rev.
19. 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.
20. P. Mitra, N. Noy, and A.R. Jaiswal. Omen: A probabilistic ontology mapping tool. In *Working Notes of the ISCW-04 Workshop on Meaning Coordination and Negotiation*, 2004.
21. 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.
22. J. Pawlak. Rough sets. *International Journal of Information and Computers*, 1982.
23. 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.
24. A. Rector. Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases. In *Proceedings of Pacific Symposium on Biocomputing*, 2004.
25. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.
26. 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.
27. 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, pages 167–181, Crete, 2005. Springer Verlag.
28. H. Stuckenschmidt and U. Visser. Semantic translation based on approximate re-classification. In *Proceedings of the 'Semantic Approximation, Granularity and Vagueness' Workshop*, 2000.
29. 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.
30. L. Zadeh. Fuzzy sets. *Information and Control*, 1965.
31. 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.
32. H.-J. Zimmermann. *Fuzzy Set Theory and its Applications*. Springer, 2001.