

## On Retrieval Guided By Extracted Domain-Specific Knowledge

**Troels Andreasen**

Department of Computer Science,  
Roskilde University  
troels@ruc.dk

**Rasmus Knappe**

Department of Computer Science,  
Roskilde University  
knappe@ruc.dk

**Henrik Bulskov**

Department of Computer Science,  
Roskilde University  
bulskov@ruc.dk

### Abstract

In this paper we introduce an approach to the modelling of conceptual similarity based on domain knowledge and an approach to aggregation to derive object similarity from concept similarity. Domain knowledge is represented in a special, so-called, domain-specific ontology, which basically is a restriction of a general ontology by a collection of domain concepts or a given document collection. Similarity is derived from the domain-specific ontology and two different variants are considered – an un-weighted and a weighted. Aggregation generalize from concept to object similarity and may be applied in text retrieval to derive answers by comparing query objects with text objects in the base.

Adopted for ontology representation is a specific lattice-based concept algebraic language by which ontologies are inherently generative.

The modelling of a domain specific ontology is based on a general ontology built upon common knowledge resources such as dictionaries and thesauri.

The resulting domain specific ontology and similarity can be applied for surveying the collection through key concepts and conceptual relations and provides a means for topic-based navigation. **Keywords:** Ontology, Information Retrieval, Fuzzy sets

### 1 Introduction

The knowledge contained in ontologies can contribute with valuable information concerning the organization of concepts, as well as the structure and relations within a given knowledge domain.

We can therefore, by incorporating ontologies into tools for information access, provide foundation for enhanced, knowledge-based approaches to surveying, indexing and querying of document collections.

We describe first the notion of a domain-specific ontology as derived from a general ontology and from concepts instantiated in a target document collection. The domain-specific ontology represents a conceptual organization reflecting a document collection and it therefore reveals domain knowledge, for instance about the thematic areas of the domain (covered by the document collection), which in turn facilitates means for access to, and querying of information, within a given domain. Secondly we introduce principles to derive similarity from domain knowledge represented in domain-specific ontologies. Lastly, we discuss approaches for validation and aggregation in connection with text retrieval applying the derived similarity.

Modelling and use of ontologies relies on the ability to identify concept occurrences in – or generate conceptual descriptions of – text. To this end we assume a processing of text by a simplified natural language parser identifying concept occurrences in the text. As this kind of processing is not the issue in this paper, we refer to [2] for a discussion of principles and for presentation of implemented parsers. It should be emphasized here, however, that when the goal is to extract descriptions that indicates semantic content, while refraining from full semantic analysis, it is possible to produce parsers that can perform efficiently also on large volumes of data.

Such a parser may be applied for indexing document as well as for interpreting queries. The most simplified principle, that was actually implemented in the project reported here, is a two-phase processing, with

the first phase being basically a noun phrase bracketing, and the second, an extract of concepts from the noun phrases individually. A naive, but useful, second phase is to extract nouns and adjectives only, and combine them into “*noun CHR adjective*”-pattern concepts (CHR representing a “characterized by” relation).

Thus, for instance, for the sentence “*the black dog is chasing the cat*” the parser may produce the following: {dog[CHR:black], cat}.

Concept expressions, that are the key to modelling and use of ontologies, are explained in more detail below.

## 2 Representation of Ontologies

The purpose of the ontology is to define and relate concepts that may appear in the document collection or in queries to this.

We define a generative ontology framework where a basis ontology situates a set of atomic term concepts  $\mathbf{A}$  in a concept inclusion lattice. A concept language (description language) defines a set of well-formed concepts, including both atomic and compound term concepts.

The concept language used here, ONTOLOG[4], defines a set of semantic relations  $\mathbf{R}$  that can be used for “attribution” (feature-attachment) of concepts to form compound concepts. The set of available relations may vary with different domains and applications. We may choose  $\mathbf{R} = \{\text{WRT, CHR, CBY, TMP, LOC, ...}\}$ , for *with respect to, characterized by, caused by, temporal, location*, respectively.

Expressions in ONTOLOG are concepts situated in the ontology formed by an algebraic lattice with concept inclusion (ISA) as the ordering relation.

Attribution of concepts – combining atomic concepts into compound concepts by attaching attributes – can be written as feature structures. Simple attribution of a concept  $c_1$  with relation  $r$  and a concept  $c_2$  is denoted  $c_1[r: c_2]$ .

Compound terms can be built from nesting, for instance,  $c_1[r_1: c_2[r_2: c_3]]$  and from multiple attribution as in  $c_1[r_1: c_2, r_2: c_3]$ .

For a formal definition of the formation rules for well-formed concepts see [1].

The attributes of a term with multiple attributes  $T = x[r_1: y_1, \dots, r_n: y_n]$  are considered as a set, thus we can rewrite T with any permutation of  $\{r_1: y_1, \dots, r_n: y_n\}$ .

## 3 Modelling Ontologies

One objective in the modelling of domain knowledge is for the domain expert or knowledge engineer to identify significant concepts in the domain.

Ontology modelling in the present context is, compared to other works within the ontology area, a limited approach. The modelling consists of two parts. Firstly an inclusion of knowledge from available knowledge sources into a general ontology and secondly a restriction to a domain-specific part of the general ontology.

### 3.1 The General Ontology

Sources for knowledge base ontologies may have various forms. Typically a taxonomy can be supplemented with for instance word and term lists as well as dictionaries for definition of vocabularies and for handling of morphology.

We will not go into details on the modelling here but just assume the presence of a taxonomy in the form of a simple taxonomic concept inclusion relation  $\text{ISA}_{\text{KB}}$  over the set of atomic concepts  $\mathbf{A}$ .  $\text{ISA}_{\text{KB}}$  and  $\mathbf{A}$  expresses the domain and world knowledge provided.  $\text{ISA}_{\text{KB}}$  is assumed to be explicitly specified – e.g. by domain experts – and would most typically not be transitively closed.

Based on  $\widehat{\text{ISA}}_{\text{KB}}$ , the transitive closure of  $\text{ISA}_{\text{KB}}$ , we can generalize into a relation  $\leq$  over all well-formed terms of the language  $\mathbf{L}$ . For a formal definition of  $\leq$  see [1].

The general ontology  $O = (\mathbf{L}, \leq, \mathbf{R})$  encompasses a set of well-formed expressions  $\mathbf{L}$  derived from the concept language, an inclusion relation generalized from an expert provided relation  $\text{ISA}_{\text{KB}}$  and a supplementary set of semantic relations  $\mathbf{R}$ . For  $r \in \mathbf{R}$  we obviously have that  $x[r: y] \leq x$  and that  $x[r: y]$  is  $x$  in relation  $r$  to  $y$ .

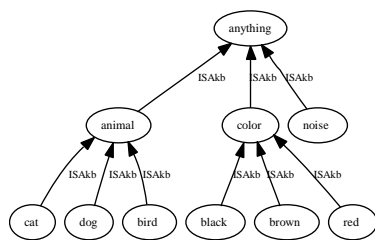


Figure 1: An example knowledge base ontology  $ISA_{KB}$

### 3.2 The Domain-Specific Ontology

Apart from the general ontology  $O$ , the target document collection contributes to the construction of the domain ontology. We assume a processing of the target document collection, where an indexing of text in documents, formed by sets of concepts from  $L$ , is attached. In broad terms the domain ontology is a restriction of the general ontology to the concepts appearing in the target document collection.

More specifically the generative ontology is, by means of concept occurrence analysis over the document collection, transformed into a domain specific ontology restricted to include only the concepts instantiated in the documents covering that particular domain.

We thus introduce the domain specific ontology as an “instantiated ontology” of the general ontology with respect to the target document collection.

The instantiated ontology  $O_{\hat{I}}$  appears from the set of all instantiated concepts  $I$ , firstly by expanding  $I$  to  $\hat{I}$  – the transitive closure of the set of terms and subterms of term in  $I$  – and secondly by producing the subontology consisting of  $\hat{I}$  connected by relations from  $O$  between elements of  $\hat{I}$ .

Consider, as an example, the knowledge base ontology  $ISA_{KB}$  shown in Figure 1. In this case we have

$$A = \{cat, dog, bird, black, brown, red, animal, color, noise, anything\}$$

and  $L$  includes  $A$  and any combination of compound terms combining elements of  $A$  with attributes from  $A$  by relations from  $R$ .

Now assume a miniature target document collection

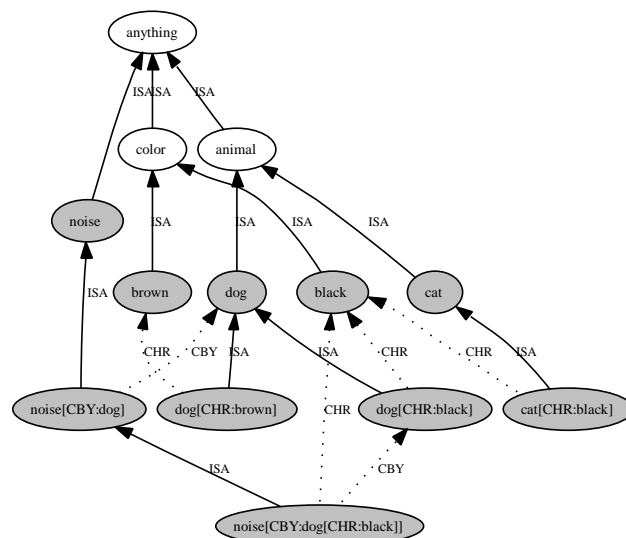


Figure 2: A simple instantiated ontology based on Figure 1 and the set of instantiated concepts  $cat[CHR:black]$ ,  $dog[CHR:black]$ ,  $dog[CHR:brown]$ ,  $noise[CBY:dog[CHR:black]]$ .

with the following instantiated concepts:

$$I = cat[CHR:black], dog[CHR:black], dog[CHR:brown], noise[CBY:dog[CHR:black]]$$

The subterms of  $I$  includes any subterm of elements from  $I$ , in this case the nodes shaded grey in Figure 2, while  $\hat{I}$  adds the subsuming  $\{animal, color, anything\}$ :

$$\hat{I} = \{cat, dog, black, brown, animal, color, noise, anything, cat[CHR:black], dog[CHR:black], dog[CHR:brown], noise[CBY:dog], noise[CBY:dog[CHR:black]]\}$$

where the concepts  $red$  and  $bird$  from  $A$  are omitted because they are not instantiated.

The resulting instantiated ontology  $(\hat{I}, \leq, R)$  is transitively reduced into the domain-specific ontology  $(\hat{I}, ISA, R)$  as shown in figure 2.

## 4 Deriving Similarity

A domain ontology, that reflects a document collection, may provide an excellent means to survey and give perspective to the collection. However as far as access to documents is concerned ontology reasoning is not the most obvious evaluation strategy and it may well entail scaling problems. Applying

measures of similarity derived from the ontology is a way to replace reasoning with simple computation still influenced by the ontology. A well-known and straightforward approach to this is the shortest path approach [5], where closeness between two concepts in the ontology imply high similarity. A problem with this approach is that multiple connections are ignored. In the ontology in figure 2 we thus have that the shortest path similarity between *cat* and *dog* would be equal to or greater than the similarity between *cat*[CHR:*black*] and *dog*[CHR:*black*] (depending on whether CHR-edges are included or not), while intuitively the former should be less than the latter because we have two concepts that meet in *animal* AND share the *black*-property.

To differentiate here an option is to consider all paths rather than only the shortest path as introduced in [1]. The similarity between two concepts  $c_1$  and  $c_2$  is the set of “upwards reachable” concepts (nodes) shared between  $c_1$  and  $c_2$ . This is, where  $\alpha(x)$  is the transitive closure of the subterms of  $x$  with respect to  $\leq$ , the intersection  $\alpha(x) \cap \alpha(y)$ .

Similarity can be defined in various ways and it is very difficult to define optimal functions both in the general and in the domain specific case. One flexible parameterized option is, as described in [3], a weighted average, where  $\rho \in [0, 1]$  determines the degree of influence of the nodes reachable from  $x$  respectively  $y$ .

$$\text{sim}(x, y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1 - \rho) \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|}$$

$\rho$  has to be obtained empirically, and can be used for tuning the similarity function towards different ontologies or domains. This similarity function is deliberately asymmetric. The motivation is the observation concerning similarity in connection with queries; a specific answer to a general question is better than a general answer to a specific question.

As it appears the upwards expansion  $\alpha(c)$  includes not only all subsuming concepts  $\{c_i \mid c \leq c_i\}$  but also concepts that appears as direct or nested attributes to  $c$  or to any subsuming concept of these attributes. The latter must be included if we want to cope with multiple connections and want to consider for instance two concepts more similar if they bear the same color.

## 5 Querying

In the present approach ontology-based querying relies on comparison of a description of the query with descriptions of texts from the database. Queries and texts are mapped onto descriptors organized in structures called “descriptions”. Processing of queries is thus facilitated by matching of descriptions in that a query answer is a ranking of the sentences with descriptions that are most similar to the description of the query.

Descriptions are not unique and may vary by level of detail, combinability and structure. Among the possible descriptions for the phrase: *The noisy black dog is chasing the cat* are the following increasingly accurate descriptions:

$$\begin{aligned} &\{noise, black, dog, cat\} \\ &\{\{noise, black, dog\}, \{cat\}\} \\ &\{\{noise, dog[chr : black]\}, \{cat\}\} \\ &\{noise[cby : dog[chr : black]], cat\} \end{aligned}$$

An approach is to index the information base with a description structure where descriptors are single concepts (as in the first and the last description example above):

$$D = \{d_1, \dots, d_n\} \quad (1)$$

A query  $Q$  may be posed in natural language and have a derived description attached in the same form,  $Q = \{q_1, \dots, q_n\}$ , or alternatively the query can be posed directly as a set of concepts (descriptors) of individual  $q_i$ .

The general idea is to capture similarity reflecting the domain-knowledge from the ontology in query evaluation, and for this purpose to use the derived similarity measures rather than to reason on the ontology. Thus a simple approach is to employ the *similar* function on either the descriptors of  $D$  or – preferably – the descriptors of  $Q$  (since we have many  $D$ 's and only one  $Q$ ).

Now the first objective is to introduce appropriate principles for similarity evaluation and for aggregation. We inspect this issue in the first subsection below, before going further in the discussion on general evaluation principles in the second.

### 5.1 Aggregation

A query  $Q$  is represented by a description  $\{q_1, \dots, q_n\}$  and we assume that the value  $q_i(D) \in [0, 1]$  is the degree to which the text object with description  $D$  satisfies the descriptor  $q_i$ . The overall valuation  $Val_Q(D)$  of object  $D$  wrt  $Q$  is obtained as an aggregation of  $\{q_1(D), \dots, q_n(D)\}$ . We adopt order weighted averaging (OWA) [6] and denote the simple OWA aggregation by the function  $F_W$ , where  $W$  is the order weighting vector. Importance weighted OWA is denoted by the function  $F_{M,W}$ , where the  $n$ -vector  $M = \langle m_1, \dots, m_n \rangle$ ,  $m_j \in [0, 1]$  are importances to  $q_1, \dots, q_n$ . We have that  $M = \langle 1, \dots, 1 \rangle$  is neutral importance and that  $F_{M,W}(q_1(D), \dots, q_n(D)) = F_W(m_1 * q_1(D), \dots, m_n * q_n(D))$ .

OWA aggregation conveniently adapts 'linguistic quantifiers', modelled by an increasing function  $K : [0, 1] \rightarrow [0, 1]$  with  $K(0) = 0$  and  $K(1) = 1$ , such that the order weights  $W$  are prescribed as:

$$w_j = K\left(\frac{j}{n}\right) - K\left(\frac{j-1}{n}\right)$$

A quantifier *EXISTS* can for instance be modeled by  $K(x) = 1$  for  $x > 0$ , *FOR-ALL* by  $K(x) = 0$  for  $x < 1$ , and *SOME* by  $K(x) = x$ , while one possibility (of many) to introduce *MOST* is by a power of *SOME*, e.g.  $K(x) = x^3$ .

Thus we have a general query expression:

$$Q = \langle q_1, \dots, q_n : M : K \rangle$$

where  $q_1, \dots, q_n$  are the query descriptors,  $M$  specifies importance weighting for these and  $K$  specifies a linguistic quantifier and thereby indicates an order weighting. The corresponding generalized valuation function is:

$$Val_Q(D) = F_{M,w(K)}(q_1(D), \dots, q_n(D)) \quad (2)$$

assuming a function  $w(K) \rightarrow [0, 1]^n$  that maps onto the set of order-weights corresponding to quantifier  $K$ .

A hierarchical approach to aggregation, generalizing OWA is introduced in [7]. Basically hierarchical aggregation extends OWA to capture nested expressions. Query attributes may be grouped for individual aggregation and the language is orthogonal in the sense that aggregated values may appear as arguments to aggregations. Thus, queries may be viewed as hierarchies. As an example we could pose a nested query expression:

$$\begin{aligned} &< q_1(D), \\ &\quad < q_2(D), q_3(D), \\ &\quad \quad < q_4(D), q_5(D), q_6(D) : M_3 : K_3 \rangle \\ &\quad \quad \quad : M_2 : K_2 \rangle, \\ &\quad \quad \quad : M_1 : K_1 \rangle \end{aligned}$$

Here again  $q_i(D) \in [0, 1]$  measures the degree to which descriptor  $q_i$  conforms to the text object with description  $D$ , while  $M_j$  and  $K_j$  are the importance and quantifier applied in the  $j$ 'th aggregate. In the expression above  $M_1 : K_1$  parameterizes aggregation at the outermost level of the two components  $q_1(D)$  and the expression in line 2 to 4.  $M_2 : K_2$  parameterizes aggregation of the three components  $q_2(D)$ ,  $q_3(D)$ , and the innermost expression (line 3), while  $M_3 : K_3$  parameterizes aggregation of the three components  $q_4(D)$ ,  $q_5(D)$ , and  $q_6(D)$ .

### 5.2 Query evaluation approaches

We distinguish two major cases of description structure – simple unested sets and nested sets.

#### 5.2.1 Aggregation on unnested descriptions

The simple set-of-descriptors structure for descriptions in (1) admits a straightforward valuation approach for a similarity query:

$$Q_{sim} = \langle q_1, \dots, q_n : (1, 1, \dots) : SOME \rangle$$

The aggregation here is simple in that importance is not distinguished and *SOME*, corresponding to simple average, is used as quantifier. A valuation can be:

$$Val_{Q_{sim}}(D) = F_{(1,1,\dots),w(SOME)}(q_1(D), \dots, q_n(D)) \quad (3)$$

with individual query-descriptor valuation functions as:

$$q_i(D) = maximum_j \{x | x/d_j \in similar(q_i)\}$$

Consider for instance the query

$$Q = \langle dog[CHR:black], noise \rangle$$

Taking a 0.4 threshold we have that

$$\begin{aligned} similar(dog[CHR:black]) = \\ &1/dog[CHR:black] + 0,7/dog[CHR:brown] + \\ &0,68/dog + 0,6/cat[CHR:black] + \\ &0,58/noise[CBY:dog[CHR:black]] + 0,52/animal + \\ &0,45/cat + 0,45/black + 0,42/noise[CBY:dog] \end{aligned}$$

$$\begin{aligned} \text{similar}(\text{noise}) = & \\ & 1,00/\text{noise} + 0,90/\text{noise}[\text{CBY:dog}] + \\ & 0,87/\text{noise}[\text{CBY:dog}[\text{CHR:black}]] + \\ & 0,60/\text{anything} + 0,50/\text{animal} + 0,50/\text{color} + \\ & 0,47/\text{cat} + 0,47/\text{black} + 0,47/\text{dog} + 0,47/\text{brown} + \\ & 0,44/\text{cat}[\text{CHR:black}] + 0,44/\text{dog}[\text{CHR:black}] + \\ & 0,44/\text{dog}[\text{CHR:brown}] \end{aligned}$$

and we get, as examples, the following:

$$\begin{aligned} \text{Val}_{Q_{sim}}(\{\text{noise}[\text{CBY:dog}]\}) &= 0.90 \\ \text{Val}_{Q_{sim}}(\{\text{noise}[\text{CBY:dog}[\text{CHR:black}]]\}) &= 0.87 \\ \text{Val}_{Q_{sim}}(\{\text{dog}, \text{noise}\}) &= 0.84 \\ \text{Val}_{Q_{sim}}(\{\text{black}, \text{dog}, \text{noise}\}) &= 0.72 \end{aligned}$$

### 5.2.2 Nested aggregation on unnested descriptions

An alternative is to expand the query  $Q$  to a nested expression:

$$\begin{aligned} \text{Val}_{Q_{sim}}(D) = & \\ & \langle \langle q_{11}(D), \dots, q_{1k_1}(D) : M_1 : K_1 \rangle, \\ & \langle q_{21}(D), \dots, q_{2k_2}(D) : M_2 : K_2 \rangle, \\ & \dots \\ & \langle q_{n1}(D), \dots, q_{nk_n}(D) : M_n : K_n \rangle, \\ & : M_0 : K_0 \rangle \end{aligned}$$

where for each  $q_i$  we set

$$\langle \mu_{i1}/q_{i1}, \dots, \mu_{ik_i}/q_{ik_i} \rangle = \text{similar}(q_i)$$

and use as individual valuation:

$$q_{ij}(D) = \begin{cases} \mu_{ij}, & \text{when } q_{ij} \in \{d_1, \dots, d_m\} \\ 0, & \text{otherwise} \end{cases}$$

In case we use equal importance and the following combination of quantifiers:

$$\begin{aligned} \text{Val}_{Q_{sim}}(D) = & \\ & \langle \langle q_{11}(D), \dots, q_{1k_1}(D) : (1, 1, \dots) : EXIST \rangle, \\ & \langle q_{21}(D), \dots, q_{2k_2}(D) : (1, 1, \dots) : EXIST \rangle, \\ & \dots \\ & \langle q_{n1}(D), \dots, q_{nk_n}(D) : (1, 1, \dots) : EXIST \rangle, \\ & : (1, 1, \dots) : SOME \rangle \end{aligned}$$

we get a valuation identical to that of (3). Nested expressions, however, facilitates importance adjustment in connections with query expansion according to the kinds of relations contributing to the expansion. Assigning 1.0 importance to ISA and 0.5 importance to CHR would, for the query  $Q = \langle \text{dog}[\text{CHR:black}], \text{noise} \rangle$ , lead to the expansion (compare with figure 2):

$$\text{Val}_{Q_{sim}}(D) =$$

$$\begin{aligned} & \langle \langle q_{\text{dog}[\text{CHR:black}]}(D), q_{\text{dog}}(D), q_{\text{black}}(D), \dots \\ & \quad : (1, 1, 0.5, \dots) : EXIST \rangle, \\ & \langle q_{\text{noise}}(D), \dots : (1, 1, \dots) : EXIST \rangle \\ & : (1, 1, \dots) : SOME \rangle \end{aligned}$$

Nested expressions is thus a way to distinguish different kinds of relations influences on similarity.

### 5.2.3 Aggregation on nested descriptions

In some cases, when text is processed by partial analysis as indicated earlier, an intrinsic structure appears as the most obvious choice for the description. The parser used in the project reported on here is a two-phase parser, grouping words in the sentence into groups corresponding to noun phrases in the first phase, and deriving compound descriptors from the words in each noun phrase individually, in the second. Thus we have as an intrinsic structure from the first phase – a set of sets (or lists) of words. Now if we always could extract a unique compound concept as descriptor from an inner set, the resulting intrinsic structure from the second phase would be the single set as assumed above. However, it is in many cases not possible, and we would therefore loose information by flattening to a single set. This suggests descriptions to be sets of sets of descriptors such that the query structure becomes:

$$\begin{aligned} Q &= \langle Q_1, \dots, Q_n \rangle \\ &= \langle \langle q_{11}, \dots, q_{1k_1} \rangle, \dots, \langle q_{n1}, \dots, q_{nk_n} \rangle \rangle \end{aligned}$$

where the  $Q_i$ 's are sets of descriptors  $q_{ij}, j = 1, \dots, k_i$ , and a text index is:

$$\begin{aligned} D &= \{D_1, \dots, D_m\} \\ &= \{\{d_{11}, \dots, d_{1l_1}\}, \dots, \{d_{m1}, \dots, d_{ml_m}\}\} \end{aligned}$$

where the  $D_i$ s are sets of descriptors  $d_{ij}, j = 1, \dots, l_i$ .

This, however, demands a modified valuation and since in this case the initial query expression is nested also a valuation over a nested aggregation becomes the obvious choice. First of all notice that the grouping of descriptors in descriptions has the obvious interpretation of a closer binding of descriptors within a group than across different groups. So we cannot individually evaluate each  $q_{ij}(D)$ , but have to compare at the level of the groups for instance by a restrictive quantification over  $q_{i1}(D_j), \dots, q_{ik_i}(D_j)$  and an *EXIST* quantification over  $j$  to get the best matching  $D_j$  for a given  $Q_i$ . A valuation can thus be:

$$\begin{aligned}
Val_{Q_{sim}}(D) = & \\
& \langle \langle \langle q_{11}(D_1), \dots, q_{1k_1}(D_1) : M_{11} : MOST \rangle, \\
& \quad \dots \\
& \quad \langle q_{n1}(D_1), \dots, q_{nk_n}(D_1) : M_{n1} : MOST \rangle \\
& : M_1 : EXIST \rangle, \\
& \quad \dots \\
& \langle \langle q_{11}(D_m), \dots, q_{1k_1}(D_m) : M_{11} : MOST \rangle, \\
& \quad \dots \\
& \quad \langle q_{n1}(D_m), \dots, q_{nk_n}(D_m) : M_{n1} : MOST \rangle \\
& : M_m : EXIST \rangle, \\
& : M_0 : SOME \rangle
\end{aligned}$$

The individual query-descriptor valuation functions can be set to:

$$q_{ij}(D_k) = \text{maximum}_i \{x | x/d_{kl} \in \text{similar}(q_{ij})\}$$

As opposed to the single set description example above, the  $q_{ij}$ 's are here the original descriptors from the query. While choices of inner quantifiers are significant for correct interpretation, the choice of *SOME* at the outer level for the component description is just one of many possible choices to reflect the users preference of overall aggregation.

## 6 Conclusion

We have introduced an approach to retrieval guided by domain-specific knowledge, extracted on the basis of a general ontology from a set of documents. The derived notion of 'domain-specific' ontology is a restriction of a general ontology with respect to the concepts instantiated in the document collection. For retrieval purposes an ontology that is specific for the actually instantiated concepts in the document collection is a means to trim the similarity retrieval processing because similar concepts taken into consideration are actually instantiated in the collection.

The derived notion of 'domain-specific' ontology is obviously not suggested as a universal approach to knowledge modelling. We cannot in general expect an ontology to be available, with an extensive coverage of world knowledge to sufficiently allow any specific domain to be modelled by a restriction. However, in many cases where domain knowledge is not available the present approach can produce valuable, useful, but rarely complete knowledge based on a set of documents. Furthermore the resulting 'domain ontology' can, apart from facilitating applications and tools for information access, give a perspective to the content of the information base. In any case closeness or similarity in the ontology plays an important role. For the

purpose of making use of this, it appears that numeric computation on similarities provides an efficient evaluation alternative to ontological reasoning - and this kind of efficiency is crucial when dealing with large volumes of data.

## References

- [1] Andreasen, T.; Bulskov, H. and Knappe, R.: On Querying Ontologies and Databases, Flexible Query Answering Systems, 6th International Conference, FQAS 2004, Lyon, France, June 24-26, 2004, Proceedings
- [2] Andreasen, T.; Jensen, P. Anker; Nilsson, J. Fischer; Paggio, P.; Pedersen, B.S.; Thomsen, H. Erdman: Content-based Text Querying with Ontological Descriptors, in Data & Knowledge Engineering 48 (2004) pp 199-219, Elsevier, 2004.
- [3] Andreasen, T., Bulskov, H., and Knappe, R.: Similarity from Conceptual Relations, pp. 179-184 in Ellen Walker (Eds.): 22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2003, Chicago, Illinois USA, July 24-26, 2003, Proceedings
- [4] Nilsson, J. Fischer: A Logico-algebraic Framework for Ontologies - ONTOLOG, in Jensen, P. Anker & Skadhauge, P. (eds.): Proceedings of the First International OntoQuery Workshop - Ontology-based interpretation of NP's, Department of Business Communication and Information Science, University of Southern Denmark, Kolding, 2001
- [5] Rada, Roy; Mili, Hafedh; Bicknell, Ellen & Blettner, Maria: Development and Application of a Metric on Semantic Nets, IEEE Transactions on Systems, Man, and Cybernetics, Volume 19, Number 1, pp. 17-30, 1989
- [6] Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making, in IEEE Transactions on Systems, Man and Cybernetics, vol 18, 1988.
- [7] Yager, R.R.: A hierarchical document retrieval language, in Information Retrieval vol 3, Issue 4, Kluwer Academic Publishers pp. 357-377, 2000.