

DOMAIN-SPECIFIC SIMILARITY AND RETRIEVAL

Troels Andreassen¹ Rasmus Knappe¹ Henrik Bulskov¹
1. Department of Computer Science, Roskilde University
Email: troels@ruc.dk, knappe@ruc.dk, bulskov@ruc.dk

ABSTRACT: In this paper we introduce an approach to the modeling 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 modeling 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 use of ontologies can contribute significantly to the organization of concepts, structure and relations within a knowledge domain.

Incorporation of ontologies in tools for information access provides foundation for enhanced, knowledge-based approaches to surveying, indexing and querying of document collections.

We introduce first in this paper 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 a topic-based navigation, visualization of the structure, and 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.

We introduce, in section 2, to a formalism for representation of ontologies. Section 3 describes the modeling of general and domain-specific ontologies, respectively. In section

4 it is discussed how to establish measures of domain-specific similarity from domain-specific ontologies. Section 5 covers querying, with emphasis on valuation and aggregation.

Both modeling and use of ontologies relies on the possibility 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 providing concept occurrences in the text. This kind of processing is not the issue in this paper but we can 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 into “*noun CHR adjective*”-pattern concepts (CHR representing a “characterized by” relation). Thus, for instance, for the sentence :

The noisy black dog is chasing the cat

the parser may produce the following:

{ noise[CBY: 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 **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[6], defines a set of semantic relations **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 $\mathbf{R} = \{\text{WRT, CHR, CBY, TMP, LOC, \dots}\}$, 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 $c_1[r: c_2]$.

Given atomic concepts \mathbf{A} and relations \mathbf{R} , the set of well-formed terms \mathbf{L} of the ONTOLOG language is defined as follows.

- if $x \in \mathbf{A}$ then $x \in \mathbf{L}$
- if $x \in \mathbf{L}$, $r_i \in \mathbf{R}$ and $y_i \in \mathbf{L}, i = 1, \dots, n$ then $x[r_1: y_1, \dots, r_n: y_n] \in \mathbf{L}$

It appears that 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 $c_1[r_1: c_2, r_2: c_3]$. 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 MODELING 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. The first part involves modeling of concepts in a generative ontology using knowledge sources WordNet [4], SUMO [5] and the American Heritage Dictionary of the English Language [www.bartleby.com]. In the second part a domain-specific ontology is retrieved as a subontology of the general ontology. The restriction to this subontology is build based on the set of concepts that appears (is instantiated) in the document collection and the result is called an instantiated 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 modeling here but just assume the presence of a taxonomy in the form of a simple taxonomic concept inclusion relation ISA_{KB} over the set of atomic concepts \mathbf{A} . ISA_{KB} and \mathbf{A} expresses the domain and world knowledge provided. ISA_{KB} is assumed to be explicitly specified – e.g. by domain experts – and would most typically not be transitively closed. Based on \widehat{ISA}_{KB} , the transitive closure of ISA_{KB} , we can generalize into a relation over all well-formed terms of the language \mathbf{L} by the following:

- if $x \widehat{ISA}_{KB} y$ then $x \leq y$
- if $x[\dots] \leq y[\dots]$ then also $x[\dots, r: z] \leq y[\dots]$, and

$$x[\dots, r: z] \leq y[\dots, r: z],$$

- if $x \leq y$ then also

$$z[\dots, r: x] \leq z[\dots, r: y]$$

where repeated \dots in each case denote zero or more attributes of the form $r_i: w_i$.

The general ontology $O = (\mathbf{L}, \leq, \mathbf{R})$ thus encompasses a set of well-formed expressions \mathbf{L} derived from the concept language from a set of atomic concepts \mathbf{A} , an inclusion relation generalized from an expert provided relation ISA_{KB} and a supplementary set of semantic relations \mathbf{R} , where for $r \in \mathbf{R}$ we obviously have that $x[r: y] \leq x$ and that $x[r: y]$ is in relation r to y . Observe that \mathbf{L} is infinite and that O thus is generative.

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 \mathbf{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} . The subterms of a term c is obtained by the decomposition $\tau(c)$. $\tau(c)$ is defined as the set of all subterms of c , which thus includes c and all attributes of subsuming concepts for c .

$$\tau(c) = \{x | c \leq x[\dots, r: y] \vee c \leq y[\dots, r: x], x, y \in \mathbf{L}, r \in \mathbf{R}\}$$

$$\tau(c) = \text{closure of } \{c\} \text{ with respect to } t$$

For a set of terms we define $\tau(C) = \bigcup_{c \in C} \tau(c)$. As an example, we have that

$$\begin{aligned} \tau(c_1[r_1: c_2[r_2: c_3]]) &= \{c_1[r_1: c_2[r_2: c_3]], c_1[r_1: c_2], c_1, \\ &= c_2[r_2: c_3], c_2, c_3\}. \end{aligned}$$

Let $\omega(C)$ for a set of terms C be the transitive closure of C with respect to \leq . Then the expansion of the set of instantiated concepts I becomes:

$$\hat{I} = \omega(\tau(I))$$

Now, the C -restriction subontology $O_C = (C, \leq, \mathbf{R})$ with respect to a given set of concepts C , is the subontology of O over concepts in C connected by \leq and \mathbf{R} . Thus the instantiated ontology $O_{\hat{I}} = (\hat{I}, \leq, \mathbf{R}) = (\omega(\tau(I)), \leq, \mathbf{R})$ is the \hat{I} -restriction subontology of O .

Finally we define ISA as the transitive reduction of \leq and consider $(\hat{I}, ISA, \mathbf{R})$ for visualization and as basis for similarity computation below.

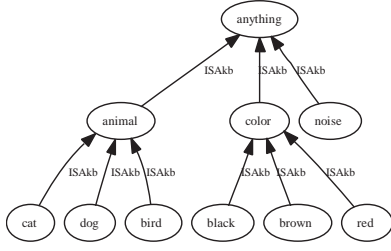


Figure 1: An example knowledge base ontology ISA_{KB}

3.3 Modeling domain-specific ontologies – An example

Consider the knowledge base ontology ISA_{KB} shown in Figure 1. In this case we have

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

and \mathbf{L} includes \mathbf{A} and any combination of compound terms combining elements of \mathbf{A} with attributes from \mathbf{A} by relations from \mathbf{R} . Now assume a miniature target document collection with the following instantiated concepts:

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

The decomposition $\tau(I)$ includes any subterm of elements from I , while $\hat{I} = \omega(\tau(I))$ adds the subsuming $\{animal, color, noise, anything\}$:

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

where the concepts *red* and *bird* from \mathbf{A} are omitted because they are not instantiated.

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

4 DERIVING SIMILARITY

As touched upon elsewhere in this paper 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 [?, 7], 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.

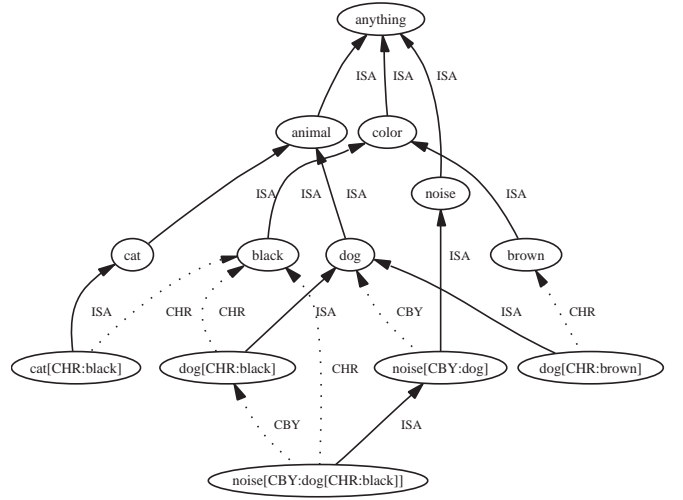


Figure 2: A simple instantiated ontology based on figure 1 and the set of instantiated concepts $cat[CHR:black]$, $dog[CHR:black]$, $dog[CHR:brown]$, $noise[CBY:dog[CHR:black]]$.

4.1 Shared nodes similarity

To differentiate here an option is to consider all paths rather than only the shortest path. A “shared nodes” approach that reflects multiple paths, but still avoids the obvious complexity of full computation of all paths is presented in [1]. In this approach the basis for 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, with $\alpha(x) = \omega(\tau(x))$, the intersection $\alpha(x) \cap \alpha(y)$.

Similarity can be defined in various ways, one option being, 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 .

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

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.

4.2 Weighted shared nodes similarity

However, a further refinement seems appropriate here. If we want two concepts to be more similar if they have an immediate subsuming concept (e.g. $cat[CHR:black]$ and $cat[CHR:brown]$ due to the subsuming *cat*) than if they only share an attribute (e.g. *black* shared by $cat[CHR:black]$ and $dog[CHR:black]$) we must differentiate and cannot just define $\alpha(c)$ as a crisp set. The following is a generalization to fuzzy set based similarity.

First of all notice that $\alpha(c)$ can be derived as follows. Let the triple (x, y, r) be the edge of type r from concept x to con-

cept y , E be the set of all edges in the ontology, and T be the top concept. Then we have:

$$\begin{aligned}\alpha(T) &= \{T\} \\ \alpha(c) &= \{c\} \cup (\cup_{(c,c_i,r) \in E} \alpha(c_i))\end{aligned}$$

A simple modification that generalizes $\alpha(c)$ to a fuzzy set is obtained through a function $weight(r)$ that attach a weight to each relation type r . With this function we can generalize to :

$$\begin{aligned}\alpha(T) &= \{1/T\} \\ \alpha(c) &= \{c\} \cup (\cup_{(c,c_i,r) \in E} weight(r) * \alpha(c_i)) \\ &= \{c\} \cup (\cup_{(c,c_i,r) \in E} \sum_{\mu(c_{ij})/c_{ij} \in \alpha(c_i)} weight(r) * \mu(c_{ij})/c_{ij})\end{aligned}$$

$\alpha(c)$ is thus the fuzzy set of nodes reachable from the concept c and modified by weights of relations $weight(r)$. For instance from the instantiated ontology in figure 2, assuming relation weights $weight(ISA) = 1$, $weight(CHR) = 0.5$ and $weight(CBY) = 0.5$, we have:

$$\begin{aligned}\alpha(noise[CBY:dog[CHR:black]]) &= \\ &1/noise[CBY:dog[CHR:black]] + 1/noise+ \\ &0.5/dog[CHR:black] + 0.5/dog + 0.5/animal + \\ &0.25/black + 0.25/color + 1/anything\end{aligned}$$

For concept similarity we can still use the parameterized expression (1) above, applying minimum for fuzzy intersection and sum for fuzzy cardinality:

$$\alpha(cat[CHR:black]) \cap \alpha(dog[CHR:black]) = 0.5/black + 0.5/color + 1/animal + 1/anything$$

$$|\alpha(cat[CHR:black]) \cap \alpha(dog[CHR:black])| = 3.0$$

The similarities between $dog[CHR:black]$ and other concepts in the ontology are, when collected in a fuzzy subset of similar concepts (with $similar(x) = \sum sim(x,y)/y$) and $\rho = \frac{4}{5}$ the following:

$$\begin{aligned}similar(dog[CHR:black]) &= \\ &1/dog[CHR:black] + 0,7/dog[CHR:brown] + \\ &0,68/dog + 0,6/cat[CHR:black] + \\ &0,58/noise[CBY:dog[CHR:black]] + 0,52/animal + \\ &0,45/cat + 0,45/black + 0,42/noise[CBY:dog] + \\ &0,39/color + 0,36/anything + ,34/brown + 0,26/noise\end{aligned}$$

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”.

When processing a query, the query description is compared, in principle, to every description of every sentence in every document appearing in the database. Finally, sentences in the documents in the database are ranked by the degree to which their description resembles the description of the query. The query answer is a ranking of the sentences that are most similar to 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):

$$D = \{d_1, \dots, d_n\} \quad (2)$$

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 principles

For similarity aggregations the class of order weighted averaging (OWA) operators [8] has shown to be very useful. These operators are especially suitable for aggregating a set of unstructured properties as the set of descriptors in the query above. The following is a brief introduction to OWA and to the more general hierarcical (or nested) aggregation.

OWA (Order Weighted Averaging) utilizes an ordering vector $W = \langle w_1, \dots, w_n \rangle$ and aggregates values a_1, \dots, a_n to:

$$F_w(a_1, \dots, a_n) = \sum_{j=1, \dots, n} w_j b_j, \quad \sum_{j=1, \dots, n} w_j = 1, w_j \in [0, 1]$$

with b_j as the j 'th largest of a_1, \dots, a_n . Thus we have b_1, \dots, b_n as the (descending) ordering of the values a_1, \dots, a_n . By modifying W we obtain different aggregations. For instance $F_{(1,0,0, \dots)}$ is maximum, $F_{(1/n, 1/n, \dots)}$ is average, and $F_{(0,0, \dots, 1)}$ is minimum.

Querying based on this may proceed from a query $\{q_1, \dots, q_n\}$ such 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 of D is:

$$Val_Q(D) = F_w(q_1(D), \dots, q_n(D)).$$

The OWA aggregation principle is very flexible and may further include importance weighting in the form of a n -vector $M = \langle m_1, \dots, m_n \rangle$, $m_j \in [0, 1]$ giving attribute importances to q_1, \dots, q_n such that for instance $M = \langle 1, 0.8, 0.8, \dots \rangle$ gives more importance to q_1 , while importances are not discriminated with $M = \langle 1, 1, \dots \rangle$. To introduce attribute importance corresponds to a modification of the valuation $val_Q(D)$ into $F_w(q_1(D) * m_1, \dots, q_n(D) * m_n)$.

In addition, the aggregation may be modelled by a 'linguistic quantifier', which basically is an increasing function

$K : [0, 1] \rightarrow [0, 1]$ with $K(0) = 0$ and $K(1) = 1$, such that the order weights are prescribed as:

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

Linguistic quantifiers lead to values of W and we can model for instance a quantifier *EXISTS* 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:

$$Val_Q(D) = F_{M,w(K)}(q_1(D), \dots, q_n(D)) \quad (3)$$

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 [9]. 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:

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

Here again $q_i(D) \in [0, 1]$ measures the degree to which attribute q_i conforms to document 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 unested descriptions

The simple set-of-descriptors structure for descriptions in (2) admits a straightforward valuation approach for a similarity query:

$$Q_{sim} = \langle q_1, \dots, q_n : (1, 1, \dots) : SOME \rangle \quad (4)$$

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:

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

with individual query-descriptor valuation functions as:

$$q_i(D) = \text{maximum}_j \{x | x/d_j \in \text{similar}(q_i)\}$$

Consider for instance the query

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

Taking a 0.4 threshold we have that

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

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

and we get for instance the following:

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

For the query

$$Q = \langle \text{noise}[\text{CBY:dog}[\text{CHR:black}]] \rangle$$

we get, again with a 0.4 threshold:

$$\begin{aligned} \text{similar}(\text{noise}[\text{CBY:dog}[\text{CHR:black}]])) = & \\ & 1,00/\text{noise}[\text{CBY:dog}[\text{CHR:black}]] + \\ & 0,73/\text{noise}[\text{CBY:dog}] + 0,52/\text{dog}[\text{CHR:black}] + \\ & 0,47/\text{noise} + 0,40/\text{dog} + 0,40/\text{black} \end{aligned}$$

and among valuations are the following:

$$\begin{aligned} Val_{Q_{sim}}(\{\text{noise}[\text{CBY:dog}[\text{CHR:black}]]\}) &= 1.00 \\ Val_{Q_{sim}}(\{\text{noise}[\text{CBY:dog}], \text{black}\}) &= 0.57 \\ Val_{Q_{sim}}(\{\text{noise}, \text{dog}[\text{CHR:black}]\}) &= 0.50 \\ Val_{Q_{sim}}(\{\text{noise}, \text{dog}, \text{black}\}) &= 0.42 \end{aligned}$$

5.2.2 Nested aggregation on unested descriptions

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

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

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:

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

we get a valuation identical to that of the function in the previous subsection. However with the nested expression we also have the option to use an unweighted similarity function and to introduce the differentiation of influence from relations by importance weighting as indicated below. For the query $Q = \langle \text{dog}[\text{CHR:black}], \text{noise} \rangle$

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

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 lose information by flattening to a single set. This indicates that a set-of-sets structure is a better description structure and suggests descriptions to be sets of sets of descriptors such that the query structure is:

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

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

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

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:

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

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

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

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 the notion of a 'domain-specific' ontology as a restriction of a general ontology to the concepts instantiated in a document collection. This 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 representative set of documents. It should also be noted that this 'automatic modelling' of domain knowledge can be performed on the basis of a set of representative concepts rather than a set of documents.

The resulting 'domain ontology' can, apart from giving a perspective to the content of the information base, be used in applications and tools for information access. For this purpose similarity measures plays an important role, providing an efficient numeric computation alternative to ontological reasoning, which in any case is needed for large volumes of data.

It should be emphasized that valuation of resemblance in general as well as modelling of similarity functions specifically is far from objective. It is not possible to define optimal functions neither in the general nor in the domain specific case. The best we can do is to specify flexible, parameterized functions on the basis of obvious 'intrinsic' properties of intuitive interpretation, and then to adjust and evaluate these functions on an empirical basis. Specifically it should be noted that

we have had good reasons for choosing the similarity function (1). However we shall not and cannot claim this function to be the best choice – only we can claim that it is efficient, has a flexible parameterization, to some extent respects the structure and relations of the ontology, and satisfies good general similarity function properties as discussed in [3].

As for the valuation functions for queries, their success of course is dependant on the fit of the similarity function, but also, and for the same reasons, the flexibility in terms of modelling with parameters is important for valuation. And especially hierarchical aggregation appears to be promising in this sense, as indicated above.

REFERENCE

- [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] Miller, George: WordNet: An On-line Lexical Database, International Journal of Lexicography, Volume 3, Number 4, 1990
- [5] Niles, I.; Pease, A.: Towards a Standard Upper Ontology, in Chris Welty and Barry Smith (eds.): Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, October 17-19, 2001.
- [6] 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
- [7] 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
- [8] Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making, in IEEE Transactions on Systems, Man and Cybernetics, vol 18, 1988.
- [9] Yager, R.R.: A hierarchical document retrieval language, in Information Retrieval vol 3, Issue 4, Kluwer Academic Publishers pp. 357–377, 2000.