

Similarity From Conceptual Relations

Troels Andreassen and Henrik Bulskov and Rasmus Knappe

Department of Computer Science

Roskilde University

P.O. Box, DK-4000, Roskilde, Denmark

{troels,bulskov,knappe}@ruc.dk

Abstract

The main focus of this paper is how to measure similarity in a content-based information retrieval environment. In the first part we define the information base, which is a generative framework where an ontology in combination with a concept language defines a set of well-formed concepts. Well-formed concepts is assumed to be the basis for an indexing of the information base in the sense that these concepts appear in descriptions attached to objects in the base. Subsequent and last we introduce an approach for measuring similarity in this framework. The measuring problem is divided into two continuous parts where we first narrow what concepts have in common, and secondly use this fragment, a similarity graph, for calculating the similarity between concepts. The purpose of narrowing or restricting what concepts have in common is to manage the generative aspect of the ontology, and to retain the greatest possible number of shared attributes and characteristics of the concepts being compared. Taking the similarity graphs as input we discuss what properties a similarity function need to satisfy to measure the degree of similarity proportional to how close the concepts are or how much they share.

1. Introduction

The objective of this paper is to devise similarity measures that utilizes knowledge from an ontology to obtain better and closer answers on a semantical level, thus comparing concepts rather than words. Better answers are primarily better ranked information base objects which in turn is a matter of better means for computing the similarity between a query and an object from the base.

The ontology plays its role behind the scenes, it defines and relates the concepts that are the basis for comparing queries and answers. However, even though it may for other reasons be relevant, it is not essential that the ontology and the concepts and relations it encloses are revealed to users.

For this reason issues on editing, browsing and visualization of the ontology become subordinate and the problem of representation of ontology can be dealt with in a different perspective.

Our claim is that when the ontology is no longer the primary base in focus, a more restrictive language with less expressive power is more suited in the present context. The main argument for this is that we can do with an incremental volume of knowledge represented in the ontology. Even very small fragments from a domain, such as a few related concepts, makes sense as an ontology if answers to queries can be improved by this. There is no need at all to insist on completeness on the coverage of a domain or a subdomain.

We consider a generative framework where an ontology in combination with a concept language defines a set of well-formed concepts. Well-formed concepts are assumed to be the basis for an indexing of the information base in the sense that these concepts appear as descriptors attached to objects in the base. Descriptors are combined in descriptions to express the semantic understanding of fragments of text, i.e. sentences. Descriptions of sentences are then again combined to form paragraphs etc.

This conceptual indexing is sought done by extraction of concepts from the information base by simple partial linguistic analysis instead of full natural language analysis. Again our claim is that the lack of completeness is not decisive because every little step from indexing by words towards indexing with concepts is a step towards conceptual indexing.

In this context, one of the major problems is to determine the similarity between the semantic elements. It is no longer only simple match of keywords in the text objects, but also the meaning of them, we have to take into consideration when we calculate the similarity between queries and objects in the base.

2. Concept Language

The purpose of the ontology is to define and relate concepts that can be used in descriptions. The ontology framework is generative in the following sense. A basis ontology defines a set of atomic concepts and situates these in a concept inclusion lattice, which basically is a taxonomy over single or multi-word concepts that are treated as atomic in the modelling of the domain. In combination with a given basis ontology, a concept language (description language) defines a set of well-formed concepts.

The concept language in focus here, ONTOLOG[1], defines a set of semantic relations which can be used for “attribution” (feature-attachment) to form compound concepts. The suitable number of available relations may vary with different domains, but among the more general relations that probably will be present in most domain modelings are WRT (With-respect-to), CHR (Characterized-by), CBY (Caused-by), TMP (Temporal), LOC (Location).

Expressions in ONTOLOG are descriptions of concepts situated in an 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 a feature structures. Simple attribution of a concept c_1 with relation r and a concept c_2 is denoted $c_1[r: c_2]$.

We assume a set of atomic concepts \mathbf{A} and a set of semantic relations \mathbf{R} , as indicated with $\mathbf{R}=\{\text{WRT, CHR, CBY, TMP, LOC, } \dots\}$. Then the set of well-formed concepts \mathbf{L} of the ONTOLOG language is recursively 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 concepts 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 multiple attributed term $T = x[r_1: y_1, \dots, r_n: y_n]$ is considered as a set, thus we can rewrite T with any permutation of $r_1: y_1, \dots, r_n: y_n$.

The basis for the ontology is a simple taxonomic concept inclusion relation ISA_{KB} , which is atomic in the sense that it defines a relation over the atomic concepts \mathbf{A} . It is considered as domain or world knowledge and may for instance express the view of a domain expert. We distinguish this (knowledge base) relation ISA_{KB} because concepts are assumed to be related by specific knowledge over the domain. For that reason we cannot expect the relation ISA_{KB} to be transitively closed or reduced and therefore define the relation ISA as the transitive closure of ISA_{KB} and the relation $\text{ISA}_{\text{REDUC}}$ as the transitive reduction of ISA_{KB} .

Based on ISA , the transitive closure of ISA_{KB} , we can generalize into a relation over all well-formed concepts of the language \mathbf{L} by the following.

- if $x \text{ ISA } y$ then $x \leq y$
- if $x[\dots] \leq y[\dots]$ then also

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

$$x[\dots, r: z] \leq y[\dots, r: z],$$
- if $x \leq y$ then also

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

where repeated \dots in each inequality denotes identical lists of zero or more attributes of the form $r_i: w_i$.

Take as an example the sentence: “*the black dog is making noise*” which can be translated into this semantic expression $\text{noise}[\text{CBY: dog}[\text{CHR: black}]]$.

Descriptions of text expressed in this language describe semantics and goes beyond simple keyword descriptions. A key question in the framework of querying is of course the definitions of similarity or nearness of terms, now that we no longer can rely on simple matching of keywords.

3. Measuring Similarity

Obviously there is no such thing as uniqueness as related to general proximity in knowledge. Moreover from just considering the potential dimensions involved it should be apparent that reasoning on similarity within concepts of knowledge can be a task of almost arbitrary complexity. So before going further into a discussion on how to measure similarity we emphasize the following. Firstly we cannot expect to find a universal measure that can be used independently of the knowledge represented and the domain in question. Rather the modest and pragmatic aim in developing measures of “conceptual similarity” is to demonstrate usefulness according to a set of preferred properties. Secondly it is essential to take computational complexity in deriving similarity into account – especially as any kind of query environment requires this. In general such environments should be capable of handling huge amounts of information base objects.

Similar concepts are concepts that have much in common. To approximate this vague notion we derive a conceptual similarity in two steps. The first step is to restrict the possible concepts to take into account as similar and the second is to derive similarity from “reasoning” within the restricted pool of concepts. We restrict by deriving the “similarity graph” and we move toward a reduced complexity by transforming formal conceptual reasoning into numerical concept similarity computation.

3.1. Similarity Graphs

A similarity graph is a subpart of the ontology represented as a graph with a subset of concepts as nodes and relations connecting these as edges. We define similarity graphs for any set of one or more concepts and specifically use the notion as a basis for similarity based on graph computations. The similarity between two concepts can thus be derived from a similarity graph covering these concepts.

Because the ontology is generative we need to be able to situate compound concepts in the ontology on the basis of the semantics of the concept. So before going into the full definition of similarity graphs we need to analyze the semantics of different forms of attribution, since a compound concept is constructed by attribution.

Consider the graph visualization of three compound concepts in Fig. 1.

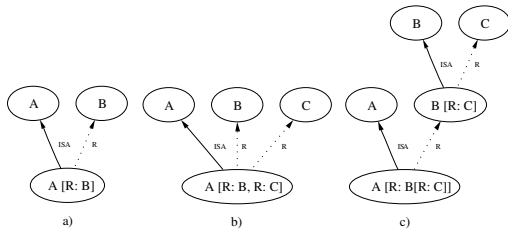


Figure 1. Visualization of three compound concepts

The leftmost concept 1a, is constructed by attribution of a concept A with a single attribute B . Attribution forms a new sub-class (specialization) of the attributed concept by adding or modifying an aspect of that concept. An example of attribution with at single attribute is the concept $car[CHR: blue]$, which obviously is a specialization of the concept car .

The centermost concept 1b is formed by attribution with a list of concepts. The semantics are the same as before, since the attribution of 1a is a special case of attribution of 1b. An example is the concept $car[CHR: blue, CHR: fast]$.

The last and rightmost concept 1c is formed by attribution with a nested attribute. Here the attribute C only modifies B and not A . Take as an example the concept $car[CHR: fast[CHR: very]]$. The “very fast car” is not a “very car”. This observation motivates the need of introducing intermediate subsuming concepts for attributes rather than just connecting direct and nested attribute atoms to the attributed atom. Thus in the car example we introduce $fast[CHR: very]$. The new concept is inserted between $car[CHR: fast[CHR: very]]$ and the concepts $fast$ and $very$.

To include in the definition of similarity graphs we first

define the term-decomposition $\tau(c)$ and the upwards expansion $\omega(c)$ of a concept term c . The term-decomposition is defined as the set of all terms appearing in c . If we for a concept $c = c_0[r_1 : c_1, \dots, r_n : c_n]$, where c_0 is the atom attributed in c and c_1, \dots, c_n are the attributes (which are atoms or further compound concepts), define:

$$subterm(c) = \{c_0, c_1, \dots, c_n\}$$

and straightforwardly extend $subterm$ to be defined on a set of concepts $C = \{c_1, \dots, c_n\}$, such that

$$subterm(C) = \cup_i subterm(c_i)$$

then we can obtain the term-decomposition of c as the closure by subterm, that is, by repeatedly applying $subterm$:

$$\tau(c) = \{c\} \cup \{x | x \in subterm^k(c) \text{ for some } k\}$$

As an example the term $noise[CBY: dog[CHR: black]]$ decomposes to the following set of concepts:

$$\begin{aligned} \tau(noise[CBY: dog[CHR: black]]) = \\ \{noise[CBY: dog[CHR: black]], \\ noise, dog[CHR: black], dog, black\} \end{aligned}$$

The upwards expansion $\omega(C)$ of a set of terms C is then the transitive closure of C with respect to ISA_{KB} .

$$\omega(C) = \{x | x \in C \vee y \in C, y ISA x\}$$

This expansion thus only adds atoms to C .

Now a similarity graph $\gamma(C)$ is defined for a set of concepts $C = \{c_1, \dots, c_n\}$ as the graph that appears when decomposing C and connecting the resulting set of terms with edges corresponding to the ISA_{KB} relation and to the semantic relations used in attribution of elements in C . We define the triple (x, y, r) as the edge of type r from concept x to concept y .

$$\begin{aligned} \gamma(C) = \cup \\ \{(x, y, ISA) | x, y \in \omega(\tau(C)), x ISA y\} \\ \{(x, y, r) | x, y \in \omega(\tau(C)), r \in \mathbf{R}, x[r: y] \in \tau(C)\} \end{aligned}$$

Fig. 2 shows an example of a similarity graph covering two terms.

3.2. Similarity

As already mentioned above we aim to derive similarity from a vaguely defined notion of how much concepts have in common. Our objective is thus to derive a function $sim(x, y)$ that measure degree of similarity proportional to how much the concepts x and y share or how close they are. Without loss of generality we assume that the function maps concepts into the unit interval:

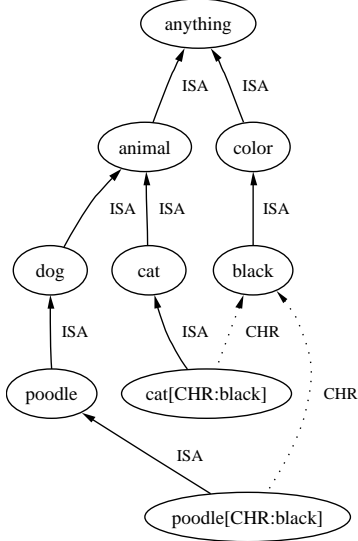


Figure 2. An example of a similarity graph for the concepts $cat[CHR: black]$ and $poodle[CHR: black]$

$$sim(x, y) : C \times C \rightarrow [0, 1]$$

where C is the set of well-formed concepts and where $sim(x, y)$ measure the degree to which y is similar to x . The extreme values $sim(x, y) = 0$ means not similar and $sim(x, y) = 1$ means fully similar. The latter may only be the case when $x = y$.

When restricting to similarity graphs one obvious approach is to reflect what connects the concepts x and y . As discussed in previous work [2] this can be done by considering the shortest path connecting the concepts x and y . As raised in [3, 4] the shortest path approach to similarity lacks the influence of an important aspect that has to do with multiple connections between concepts. We may for instance have concepts connected directly through inclusion and in addition through an attribute dimension, as $cat[CHR : black]$ and $poodle[CHR : black]$. Taking all possible paths connecting two concepts x and y solves this problem, but involves a substantial increase in complexity. If we can reflect the multiple connections phenomenon without traversing all possible paths we may have a more realistic means of similarity derivation. One option in this direction is to put emphasis on the nodes “shared” by x and y .

With $\alpha(x)$ as the set of nodes (upwards) reachable from x , thus $\alpha(x) = \omega(\tau(x))$, we have $\alpha(x) \cap \alpha(y)$ as the reachable nodes shared by x and y , which thus obviously is an indication of what’s common between x and y . Immediate transformations of this into a normalized similarity measure

are the fractions of the cardinality of the intersection and the cardinality of respectively the union $\alpha(x) \cup \alpha(y)$ and the individual $\alpha(x)$ and $\alpha(y)$ thus the following normalized measures:

- (a)
$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x) \cup \alpha(y)|}$$
- (b)
$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|}$$
- (c)
$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|}$$

There is no question about that we by similarity graphs and functions as the above only obtain a very coarse-grained approximation of whatever genuine similarity may be. However the fact that the similarity is coarse is in itself typically an advantage rather than a problem in connection with querying, if only the measuring respects or “goes in the same direction as” the semantics. In the discussion of similarity functions below we attempt to encircle major properties that ensure a given functions accordance with the semantics of the ontology and use these to guide the choice of function.

First of all it is important to notice that the similarity graph principle unifies the concept inclusion relation with the semantic relations used in attribution. We still consider upwards and downwards in the unified graph as generalization and specialization respectively, but it is important to notice that this is no longer strictly subsumption based. We thus take not only cat but also $black$ to be “generalizations” of $cat[CHR : black]$ and even $cat[CHR : black]$ to be a generalization of $accident[CBY : cat[CHR : black]]$.

A major property to guide the choice of similarity function is:

Generalization cost property

the “cost” of generalization should be significantly higher than the cost of specialization.

The intuition being that for instance a “cat” satisfies the intention of an “animal” while an “animal” (that could be of any kind) not necessarily satisfies the intention of a “cat”. From this property alone we can eliminate the first alternative similarity function (a) above. The consequence of insisting on this property is namely that the similarity function cannot be symmetrical, which (a) obviously is. In fig. 3, for instance, the (a) alternative similarity function gives $sim(D, E) = sim(E, D) = \frac{2}{5}$, while we should have $sim(D, E) < sim(E, D)$ according to the generalization cost property.

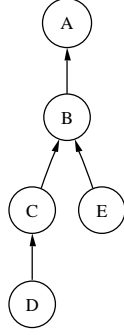


Figure 3. Generalization cost property implies $sim(D,E) < sim(E,D)$ and Specialization cost property implies $sim(C,E) > sim(D,E)$

Now consider alternative (c). We get in fig. 3 that $sim(D,E) = \frac{2}{3}$ and $sim(E,D) = \frac{2}{4}$, which also violates the specialization cost property. Thus the only alternative that obey the property is (b). With the example in fig. 3 we get $sim(D,E) = \frac{2}{4}$ and $sim(E,D) = \frac{2}{3}$

A second property that tends to appear more as optional since it is not implied by the semantics of the ontology is the following:

Specificity cost property
the "cost" of traversing edges should be lower when nodes are more specific.

The intuition for this property is that the similarity between for instance siblings on low levels in the ontology as "alsatian" and "puddle" should be higher than the similarity between siblings close to the top as "Physical" and "Abstract".

Thus in fig. 4 we should have that $sim(C,D) > sim(A,B)$. The similarity function (b) above appears to satisfy this property: we have $sim(A,B) = \frac{2}{3}$ while $sim(C,D) = \frac{4}{5}$. (Also (a) and (c) satisfies this property.)

A third property that similarly cannot be claimed to be semantically implied is the following.

Specialization cost property
further specialization implies reduced similarity.

As support of the intuition for this property consider again fig. 3. The (b) similarity function does obviously not satisfy this property since we have $sim(E,C) = sim(E,D)$ and for K at any level of specialization below D we still have $sim(E,D) = sim(E,K)$.

This motivates to consider alternative similarity functions that are influenced by both specializations and generalizations (as the function (a) above is), but still not violates the only ultimate property above; the Generalization cost

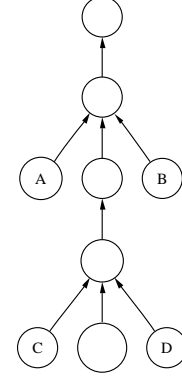


Figure 4. Specificity cost property: implies that $sim(C,D) > sim(A,B)$.

property (the anti-symmetry requirement). One modification that satisfies this is to simply take a weighted average of (b) and (c) above as the following:

(d)

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

where $\rho \in [0, 1]$ determines the degree of influence of generalizations.

Although simplicity is in favor of similarity (b) and from the aspects discussed this measure cannot be claimed to violate the semantics of the ontology, similarity (d) still appears to be a better choice. (b) is just a special case of (d) with $\rho = 1$, the parameter ρ allows to tailor the similarity function, and in general with $\rho > 0$ (d) also complies with the generalization property.

Example

As illustration of how (b) and (d) differs consider the subontology in fig. 2. The similarities for *poodle* respectively *poodle[CHR : black]* and the other concepts included in the subontology are, when collected in fuzzy subsets of similar concepts (with $similar(x) = \sum sim(x,y)/y$) are the following.

For (b) we get

$$\begin{aligned} similar(poodle) = & 1.00/poodle + 1/poodle[CHR : black] + \\ & 0,75/dog + 0,5/animal + 0,5/cat + \\ & 0,5/cat[CHR : black] \end{aligned}$$

$$\begin{aligned} similar(poodle[CHR : black]) = & 1.00/poodle[CHR : black] + 0,57/poodle + \\ & 0,57/cat[CHR : black] + 0,43/dog + \end{aligned}$$

$$0,43/black + 0,29/animal + 0,29/cat + 0,29/color$$

and (d) with $\rho = \frac{1}{3}$ leads to

$$\begin{aligned} similar(poodle) = & \\ & 1.00/poodle + 0,92/dog + 0,83/animal + \\ & 0,71/poodle[CHR : black] + 0,61/cat + \\ & 0,36/cat[CHR : black] \end{aligned}$$

$$\begin{aligned} similar(poodle[CHR : black]) = & \\ & 1.00/poodle[CHR : black] + 0,86/poodle + \\ & 0,81/dog + 0,81/black + 0,76/animal + \\ & 0,76/color + 0,63/cat[CHR : black] + \\ & 0,54/cat \end{aligned}$$

4. Conclusion

We have described different principles for measuring similarity that tries to manage the generative nature of the ontology, while still taking the greatest possible number of attributes shared by the concepts being compared.

The notion of similarity graphs as the basis for similarity measures based on shared nodes seems to indicate a usable theoretical foundation for design of conceptual similarity measures.

The purpose of similarity measures in connection with querying is of course to look for similar rather than for exactly matching values, that is, to introduce soft rather than crisp evaluation. As indicated through examples above one approach to introduce similar values is to expand crisp values into fuzzy sets including also similar values. Expansion of this kind, applying similarity based on knowledge in the knowledge base, is a simplification replacing direct reasoning over the knowledge base during query evaluation. The graded similarity is the obvious means to make expansion a useful - by using simple threshold values for similarity the size of the answer can be fully controlled.

While not the key issue here, we should point out the importance of applying an appropriate averaging aggregation when comparing descriptions. Approaches to aggregation that can be tailored to obtain the necessary properties, based on order weighted averaging[7] and capturing nested structuring[8], are described in [5, 6].

A facet for further investigation is the nature of synonyms. Synonyms are ultimately similar from a semantical point of view - an aspect that is not reflected in counting their shared nodes.

Acknowledgments

The work described in this paper is part of the OntoQuery¹ project supported by the Danish Technical Research

¹The project has the following participating institutions: Centre for Language Technology, The Technical University of Denmark, Copenhagen

Council and the Danish IT University.

References

- [1] 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.
- [2] Bulskov, H., Knappe, R. and Andreasen, T.: On Measuring Similarity for Conceptual Querying, LNAI 2522, pp. 100-111 in T. Andreasen, A. Motro, H. Christiansen, H.L. Larsen (Eds.): Flexible Query Answering Systems 5th International Conference, FQAS 2002. Copenhagen, Denmark, October 27-29, 2002. Proceedings
- [3] Knappe, R., Bulskov, H. and Andreasen, T.: On Similarity Measures for Content-based Querying, LNAI, to appear in International Fuzzy Systems Association, World Congress, June 29-July 2, Istanbul, Turkey, 2003, Proceedings
- [4] Andreasen, T., Bulskov, H. and Knappe, R.: On ontology-based querying, to appear in Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico, Proceedings
- [5] Andreasen, T.: On knowledge-guided fuzzy aggregation. In *IPMU'2002, 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1-5 July 2002, Nancy, France
- [6] Andreasen, T.: Query evaluation based on domain-specific ontologies. In *NAFIPS'2001, 20th IFSA / NAFIPS International Conference Fuzziness and Soft Computing*, pp. 1844-1849, Vancouver, Canada, 2001.
- [7] Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making, in *IEEE Transactions on Systems, Man and Cybernetics*, vol 18, 1988.
- [8] Yager, R.R.: A hierarchical document retrieval language, in *Information Retrieval* vol 3, Issue 4, Kluwer Academic Publishers pp. 357-377, 2000.

Business School, Roskilde University, and the University of Southern Denmark.