OBJECT SIMILARITY IN ONTOLOGIES: A FOUNDATION FOR BUSINESS INTELLIGENCE SYSTEMS AND HIGH-PERFORMANCE RETRIEVAL

Abraham Bernstein
Department of Informatics
University of Zurich
Zurich, Switzerland
bernstein@ifi.unizh.ch

Esther Kaufmann
Department of Informatics
University of Zurich
Zurich, Switzerland
kaufmann@ifi.unizh.ch

Christoph Buerki
Department of Informatics
University of Zurich
Zurich, Switzerland
christoph.buerki@ubs.com

Mark Klein
Center for Coordination Science
Massachusetts Institute of Technology
Cambridge, MA U.S.A.
m_klein@MIT.EDU

Abstract

Finding good algorithms for assessing the similarity of complex objects in ontologies is central to the functioning of techniques such as retrieval, matchmaking, clustering, data-mining, semantic sense disambiguation, ontology translations, and simple object comparisons. These techniques provide the basis for supporting a wide variety of business intelligence computing tasks like finding a process in a best practice repository, finding a suitable service provider or outsourcing partner for agile process enactment, dynamic customer segmentation, semantic web applications, and systems integration. To our knowledge, however, there exists no study that systematically compares the prediction quality of ontology-based similarity measures. This paper assembles a catalogue of ontology-based similarity measures that are (partially) adapted from related domains. These measures are compared to each other within a large, real-world best practice ontology as well as evaluated in a realistic business process retrieval scenario. We find that different similarity algorithms reflect different notions of similarity. We also show how a combination of similarity measures can be used to improve both precision and recall of an ontology-based, query-by-example style, object-retrieval approach. Combining the study's findings with the literature, we argue for the need of extended studies to assemble a more complete catalogue of object similarity measures that can be evaluated in many applications and ontologies.

Keywords: Semantic similarity measures, ontology, object retrieval, semantic web, business intelligence, best practice repositories, process ontologies

Introduction

Susan is a consultant who just finished an intensive brainstorming session with a customer discussing a newly redesigned sales process. From her experience, Susan knows that analogies from other industries and companies oftentimes help to find unexpected solutions to a process design problem. Therefore, she decides to look up the relevant information in a (business process) best practice repository, which is organized as a big ontology. She formulates a query specifying a number of attributes and features of her process and some of its relationships. When executing the query, however, she is buried in hundreds of results…

This is a very typical situation. People querying databases oftentimes find themselves either *buried in results* to their queries or find *no results whatsoever*. A common approach to dealing with these problems is to rank the results of a query, in the case of too many answers, or return similar returns, when no precise matches to the query exist (Baeza-Yates and Ribeiro-Neto 1999; Brin and Page 1998). Both of these approaches require a measure of *similarity* between queries and answers. Finding a good measure of similarity is, thus, crucial for providing a good retrieval performance. But not only retrieval of objects profits from a good similarity measure. A variety of techniques, such as clustering, data-mining, semantic sense disambiguation, ontology translation, automatic database schema matching, and object comparison rely on good similarity measures. Furthermore, all of these techniques have a huge impact on practical business problems such as finding a process in a best practice repository, finding a suitable service provider or outsourcing partner for agile process enactment, dynamic customer segmentation, and systems integration. As a consequence, similarity prediction algorithms are a central element in the semantic web, artificial intelligence, or computer science researcher's toolbox in order to build useful applications for everyday business use.

The increased use of ontologies for determining the semantic meaning of data raises the question of an appropriate similarity measure for the use with ontologies or semantically enhanced applications. Most semantic-web systems, however, use traditional logic approaches where corresponding objects are determined by perfect matches and similarity (as opposed to equivalence or subsumption) isn't used as a concept. Humans, on the other hand, typically have little difficulty in determining the intended meaning of ambiguous words, expressions, or even complex objects, whereas it is challenging to replicate this process computationally. This paper investigates algorithms for determining the semantic similarity between complex (i.e., aggregated or compound) objects in an ontology, as there is reason to believe that the word-vector based similarity measures traditionally used in information retrieval are unsuitable for this purpose. In particular, it *experimentally evaluates a number of adapted or existing computational measures* (mainly taken from the computer linguistics and natural language processing domain) *in a practical use scenario within* a realistic application.

As such, the contributions of this paper are the following: First, it assembles a catalogue of *similarity measures for the use in ontologies of complex objects by adapting measures from related domains* such as natural language processing (NLP). Second, it *compares the measures with each other in a large, real-world process ontology* (with over 5,000 entries) finding, among other things, that different similarity algorithms reflect *different notions of similarity*. Third, the use of the similarity algorithms is evaluated within a practical process retrieval scenario, which shows how they can be used to *improve recall and precision of an object retrieval query* by extending the reach of a logically specified query with the use of similarity measures as well as ordering the returns of such a query. In addition, we will see how suitable the different similarity algorithms are for that purpose. Last but not least, we show that such a similarity-extended logic retrieval approach is superior to the pure keyword one in both precision and recall.

The paper is structured as follows: Next, we review the literature on object similarity and present the findings as a catalogue of ontology-based object similarity functions. Then, we provide a detailed explanation of our evaluation setup, present the results of the evaluations, and discuss limitations of the presented study. We close with a discussion of related and future work.

Semantic Similarity

The question of similarity is a heavily researched subject in the computer science, artificial intelligence, psychology, and linguistics literature. In particular the information retrieval literature has a long tradition of looking at measures for the similarity between *documents* (Baeza-Yates and Ribeiro-Neto 1999; Salton and McGill 1983). Those approaches typically take the single words (or word stems) of a document as features and operate on histogram-vectors thereof, usually ignoring the ontological relationships of the words.

There are essentially two ways to make use of the hierarchical ontology structure for determining the *semantic similarity between objects in an ontology*: the *edge-based* approach and the *node-based* approach. The traditional edge-based approach estimates the distance or edge length between nodes (Lee et al. 1993; Rada et al. 1989). The shorter the path from one node to the other, the more similar they are. The problem with this approach is that it relies on the notion that edges in a taxonomy represent uniform distances (i.e., it assumes that all semantic links are of equal weight). The newer node-based approaches (Resnik 1995; Resnik 1999) typically use information content measures or information about object-part relationships to determine the conceptual similarity. The similarity between concepts is determined by the extent to which they share information.

In this section we will present five different similarity measures, both node and edge-based, that are derived from the literature and are adapted to the context of comparing *complex objects in an ontology*. As complex objects we define entities with attributes, attribute values, and relationships of which one might be a specialization (i.e., an *is-a* relationship denoting any type of subclassing). This definition subsumes both explicit ontologies such as WordNet (Miller et al. 1993), where the specialization relationship is explicitly defined using an explicit relationship, as well as ontologies, where this relationship is to be derived logically (e.g., using subsumption [Baader et al. 2003]). Consequently, we consider complex objects such as classes and instances in a semantic web ontology or a programming language, entities and records in a relational or object-oriented database, as well as any other compound data structure. For each of the measures we will briefly explain its source and how its adaptation to complex objects works.

Ontology Distance

The most intuitive similarity measure of objects in an ontology is their distance within the ontology. Obviously, *sparrows* are more similar to *geese* than to *whales*. They also reside closer in the typical biological taxonomies. The calculation of the ontology distance is based on the specialization graph of objects in an ontology. The graph representing a multiple inheritance framework is not a tree but a directed acyclic graph. In such a graph the ontology distance could be defined as the shortest path going through a common ancestor or as the general shortest path, potentially connecting two objects through common descendants or specializations. For the purposes of this study we decided to employ the former, common-ancestor based specification, which seems to better reflect the common sense understanding of the closeness of two objects in a taxonomy. The pseudo-code algorithm looks as follows:

1. gen_a = all transitive generalizations of the object A
2. gen_b = all transitive generalizations of the object B
3. from gen_a ∩ gen_b determine the most recent common ancestor (MRCA)
4. ontology distance = count the length of the path from A to MRCA to B

Information-Theoretic Approaches

The problem of the ontology distance is that it is highly dependent on the construction of the ontology. The measure is, therefore, highly dependent on oftentimes subjective ontology engineering decisions. To address this problem, researchers in the NLP domain have proposed measuring the similarity between two objects (in their case, words) in an ontology (i.e., WordNet) in terms of information-theoretic entropy measures (Lin 1998; Resnik 1999). Specifically, Resnik (1995, 1999) argues that an object (i.e., word) is defined by the members of the class specified. When using an explicit ontology like WordNet, the set of members is equivalent to the descendants (hyponyms) of an object (word). The information of a class is defined as the probability *P(.)* of finding the particular set of descendants, its entropy as the negative log of that probability. Similarity is now defined as

$$\mathrm{sim}(A,B) = (\ 2*\log P(\mathrm{MRCA}(A,B))\ )\ /\ (\ \log P(A) + \log P(B)\ ), \tag{1}$$

where MRCA is the most recent common ancestor of classes A and B. Intuitively, this measure specifies similarity as the probabilistic degree of overlap of descendants between two objects. Modeling his evaluation on an experiment by Miller and Charles (1991), which uses human subjects to rate the similarity between 30 noun pairs, Resnik shows that this information theory based method provides significant improvement (correlation 0.79) over traditional edge methods (correlation 0.60) when used in WordNet. We can directly reuse this approach for complex objects resulting in the following algorithm:

1. U = the total number of objects
2. Find the most recent common ancestor (MRCA) of A and B
3. P(A) = (number of specializations of A) / U
4. P(B) = (number of specializations of B) / U
5. P(MRCA) = (number of specializations of MRCA) / U
6. sim(A,B) = (2*log P(MRCA(A,B))) / (log P(A) + log P(B))

Vector Space Approaches

Vector space models are very common in information retrieval (Baeza-Yates and Ribeiro-Neto 1999; Salton and McGill 1983) and machine learning (Mitchell 1997). They represent each object as a vector of features in a k-dimensional space and compute the similarity by measures such as cosine or Euclidean distance. We adapted the vector space model to the complex object setting by representing it as a k-dimensional vector. Here k is the number of unique object attributes or relations (having the same value) of the object and the length of the k^{th} component of the vector is associated with the object part frequency in the objects. The similarity between two objects' vectors is now simply defined as their inner product. The pseudo-code algorithm is:

1. Determine vector *x* from the object parts of A
2. Determine vector *y* from the object parts of B
3. sim(A,B) = $|xy| / (|x| * |y|)$

As an example, consider the object *chair*, which has four *legs* and one *back* to which it has a *has-part* relation as well as a room *office* to which it has a *is-in* relation. The chair vector [4, 1, 1] would represent the chair in the space with the dimensions [*has-part_legs*, *has-part_back*, *is-in_office*]. Clearly, this type of "vectorization" is problematic as it, for example, does not capture that the dimensions *has-part_legs* and *has-part_back* are (semantically) closer related to each other than to *is-in_office*. However, it has the advantage of being computationally cheap. We, therefore, decided to use this measure as one option out of a whole set of possible vectorizations. An exhaustive study of complex object similarity measures would have to consider other vector space encodings as they are investigated in the propositionalization of relational machine learning problems (Dzeroski and Lavrac 2001) and is beyond the scope of this paper.

Edit Distance (Levenshtein Distance)

The similarity between strings is often described as the edit distance (also called the Levenshtein distance [Levenshtein 1966]), the number of changes necessary to turn one string into another. Here a change is typically defined as either the insertion of a symbol, the removal of a symbol, or the replacement of one symbol with another. In our case, we do not need to compare strings but objects. Therefore, we calculate the number of transformation steps needed to turn one object into another object. In other words, we count the number of insert, remove, and replacement operations of attributes, attribute values, relationships, or relationship types. In a first version we assume equal costs (=1) for each of the transformations. In an alternative implementation, we weigh each transformation type with a value that represents the "real" costs. For example, is the replacement transformation comparable with a deletion procedure followed by an insertion procedure? Hence, we could argue that the cost function c would have the following behavior:

$$c(\text{deleting}) + c(\text{inserting}) >= c(\text{replacing}) \tag{2}$$

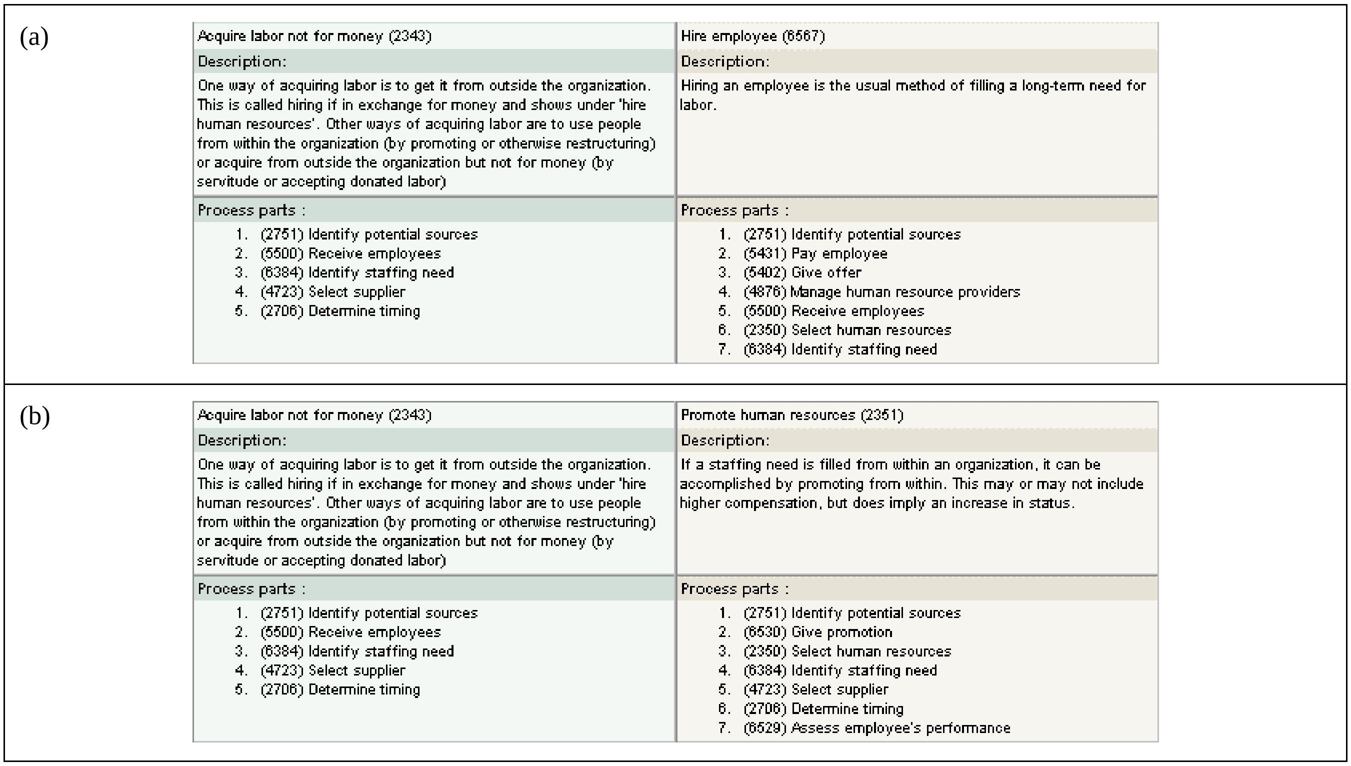
Using this assumption we calculate the worst (i.e., most costly) case for a transformation from A to B by replacing all object parts of A with object parts of B, then deleting the rest of the object parts of A, and inserting additional object parts of B. The worst case cost is then used to normalize the edit distance to a similarity. The overall algorithm looks as follows:

1. Determine parts (attributes/relationships) of A
2. Determine parts of B
3. Compute *number of transformation steps* (replace, insert, delete) from A to B
4. Compute *worst case cost* for the procedure
5. Relative edit distance = (number of transformation steps) / (worst case costs)

Full-Text Retrieval Method

Probably the most often-used similarity measure comes from the information retrieval literature and compares two documents by using a weighted histogram of the words they contain (Baeza-Yates and Ribeiro-Neto 1999; Salton and McGill 1983). Specifically, the *term frequency and inverse document frequency* measure (short TFIDF) works as follows: it counts the frequency of occurrence of each term in a document in relation to the term's occurrence frequency in a whole corpus of documents. The resulting word counts are then used to compose a weighted term vector describing the document. The similarity between two documents is now computed as the cosine between their respective weighted term vectors.

In our case we created a (text) document for each object in the ontology. Every document contained the object name, its attributes, and a brief description of its relationships (similar to the descriptions shown in Figure 1). We then used an off-the-shelf algorithm to compute the similarity of these documents (McCallum 1996).



(a)

Acquire labor not for money (2343)	Hire employee (6567)
Description:	Description:
One way of acquiring labor is to get it from outside the organization. This is called hiring if in exchange for money and shows under 'hire human resources'. Other ways of acquiring labor are to use people from within the organization (by promoting or otherwise restructuring) or acquire from outside the organization but not for money (by servitude or accepting donated labor)	Hiring an employee is the usual method of filling a long-term need for labor.
Process parts :	Process parts :
1. (2751) Identify potential sources 2. (5500) Receive employees 3. (6384) Identify staffing need 4. (4723) Select supplier 5. (2706) Determine timing	1. (2751) Identify potential sources 2. (5431) Pay employee 3. (5402) Give offer 4. (4876) Manage human resource providers 5. (5500) Receive employees 6. (2350) Select human resources 7. (6384) Identify staffing need

(b)

Acquire labor not for money (2343)	Promote human resources (2351)
Description:	Description:
One way of acquiring labor is to get it from outside the organization. This is called hiring if in exchange for money and shows under 'hire human resources'. Other ways of acquiring labor are to use people from within the organization (by promoting or otherwise restructuring) or acquire from outside the organization but not for money (by servitude or accepting donated labor)	If a staffing need is filled from within an organization, it can be accomplished by promoting from within. This may or may not include higher compensation, but does imply an increase in status.
Process parts :	Process parts :
1. (2751) Identify potential sources 2. (5500) Receive employees 3. (6384) Identify staffing need 4. (4723) Select supplier 5. (2706) Determine timing	1. (2751) Identify potential sources 2. (6530) Give promotion 3. (2350) Select human resources 4. (6384) Identify staffing need 5. (4723) Select supplier 6. (2706) Determine timing 7. (6529) Assess employee's performance

Figure 1. Two Example Process Pairs

Application-Oriented Evaluation

The similarity measures introduced above provide a first catalogue of candidates for an ontology-based similarity metric. All of them have been used in some form or another in other domains and, therefore, have the potential of being useful in the semantic-web domain. In order to assess their usefulness, however, we need to compare the measures in the context of a real-world ontology and evaluate them in an application for similarity measures.[1] To that end we designed both a statistically sound comparison and an ontology retrieval experiment, which we will discuss in the remainder of this section.

Similarity Measure Comparison in a Real-World Ontology

To evaluate the similarity measures, we chose object pairs from an ontology and looked how the various measures correlated. As the *underlying ontology we chose the MIT Process Handbook ontology* (Malone et al. 2003; Malone et al. 1999), which contains over 5,000 organizational processes and has been carefully developed for over 10 years. The ontology has a number of advantages. It treats a real-world usage domain to which everybody can relate. Each process in the ontology has a variety of relationships to attributes, subprocesses, exceptions, etc., and also has a detailed textual description, providing multiple types of information about the processes. Finally, the ontology has been used (and validated) in many projects (Malone et al. 2003) and treats a domain of interest to information systems researchers. Unfortunately, the Process Handbook is sometimes confusing in that it, like WordNet, doesn't distinguish between instances and classes. We selected 20 process pairs from the Process Handbook. The process pairs were chosen semi-randomly. In other words we restricted the choice as follows: All chosen processes should have at least one specialization in order to allow the information-theoretic similarity measure to work. Furthermore, we ensured that at least some of the pairs would have ancestor/descendant relationships with each other.

[1] Alternatively, we could have evaluated the measures' usefulness against a "gold standard" of object similarity, which is beyond the scope of this paper. Such an evaluation is typical in the NLP domain (Budanitsky and Hirst 2001; Jarmasz and Szpakowicz 2001), where the focus lies on word similarity rather than object similarity.

As an example consider the process pairs in Figure 1. Both pairs show similar processes in the domain of human resources. The process pair (*a*) compares the process of hiring employees (i.e., work for money) with the process of "acquiring"[2] labor not for money (e.g., finding volunteer workers). Note that they share some subprocesses, such as "identify potential sources," while others (such as "pay employee") are unique to one of the processes. The process pair (*b*) compares the "acquisition" of labor with the promotion within an organization. All three processes are also shown in Figure 2, which depicts an excerpt of the specialization relationships within the Process Handbook ontology, where the relevant processes are in boldface (and quite a few processes are hidden). Given this information about the ontology, we can compute the similarities between the process pairs. The absolute edit distance in both process pairs is the same (4). The vector similarity is slightly higher for the pair *b)* (0.676) than for the pair *a)* (0.507). This correlates with the similarity assessment of the information theory finding (*b*): 0.867; (*a*): 0.542) and the ontology distance (*b*): 2; (*a*): 8).[3] These similarity assessments seem intuitively correct, as they reflect the absence of money in the bottom vs. the top pair.

To assess the quality of the similarity algorithms beyond the example given above, we compared their assessments for each of the chosen process pairs. This turned out to be a non-trivial task. First, some algorithms provided nominal predictions; others generate assessments on an ordinal scale. Second, the prediction of some algorithms was nonlinear, complicating their comparison using traditional correlations. We, therefore, *compared each pair of assessments using the corrected Spearman correlation coefficient r_s*, which compares bindings (corrected ranks) of assessments rather than absolute values addressing both issues (Sachs 2002). This coefficient compares two paired sets by assigning each number a binding rank with respect to its set and provides a number r_s between –1 and 1, where 1 represents perfectly correlated sets, –1 inversely correlated sets, and 0 completely uncorrelated sets. Typically values of $r_s \geq 0.5$, respectively $r_s \leq -0.5$, are taken as some indication of (respectively, inverse) correlations and values of $r_s \geq 0.6$, respectively $r_s \leq -0.6$, as good indication of correlations.[4] In other words, we took each series of similarity assessment and compared it to every other assessment using the corrected Spearman rank correlation.

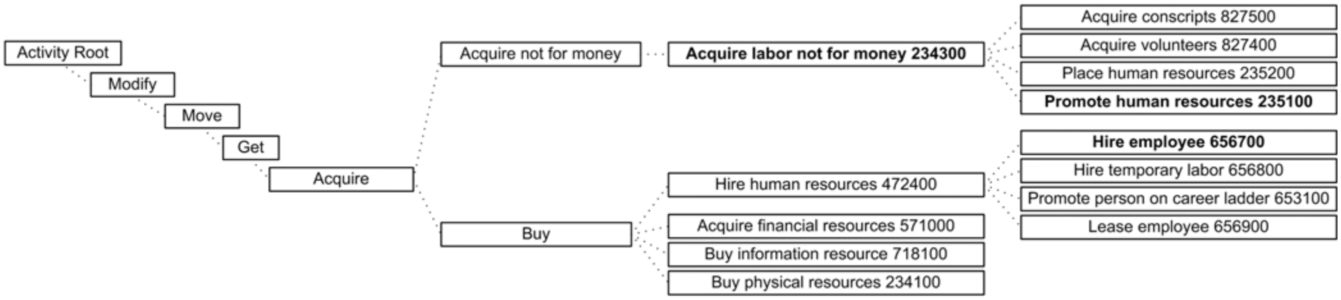

Activity Root
Modify
Move
Get
Acquire
Acquire not for money
Buy
Acquire labor not for money 234300
Hire human resources 472400
Acquire financial resources 571000
Buy information resource 718100
Buy physical resources 234100
Acquire conscripts 827500
Acquire volunteers 827400
Place human resources 235200
Promote human resources 235100
Hire employee 656700
Hire temporary labor 656800
Promote person on career ladder 653100
Lease employee 656900

Figure 2. Ontology Excerpt Containing Specialization Relationships of a Partial View of the "Acquire" Branch
(Note that many elements have been removed to allow the inclusion of the figure)

[2] We use the names given in the ontology, which don't necessarily represent our understanding but the ontology's.

[3] These numbers cannot be inferred directly from Figure 2, as some special ontology elements called "bundles" in the Process Handbook are omitted in the figure. More information on bundles can be found in Malone et al. 2003.

[4] The exact cutoffs for the significance of the relationship is dependent on the degrees of freedom (number of ranked items minus 2) compared (Sachs 2002).

Table 1. Corrected Spearman Rank Correlations between the Similarity Assessment Methods

Algorithm	O	I	V	E	W	T
Ontology Distance (O)	1	–0.93**	–0.70**	0.49	0.65*	–0.58*
Information Theory (I)		1	0.60^{+}	–0.37	0.65*	0.57*
Vector Space (V)			1	–0.91**	–0.99**	0.75**
Edit Distance (E)				1	0.95**	–0.64*
Weighted ED (W)					1	–0.7**
TFIDF (T)						1

**p = 0.001; *p = 0.01

Table 1 shows the resulting coefficients. We can discern two groups of methods. The first group, consisting of the ontology distance and the information theory has an absolute r_s of 0.93 (i.e., its similarity predictions correlate highly). The second group, consisting of the two edit distances, the TFIDF measure, and the vector space model,[5] has an average absolute r_s of 0.82 (or 0.95 without TFIDF). Both groups seem to have an underlying structure. *The first group of measures, the information-theoretic approach and the ontology distance, are oriented toward the location of a process in the ontology*: the information-theoretic approach through its reliance on the processes' descendants, which are likely to be more common for closer objects, and the ontology distance by the direct count of the closeness in the ontology. Also, these two measures did correlate in the NLP experiments (Resnik 1999). *The second group of measures, the two edit distances and the vector model, largely focus on the objects' parts*, i.e., their attributes and relationships (especially the subprocesses). The correlation between the two edit distances is to be expected; essentially they only differ in the weights. The vector space model is highly similar in that it builds a vector from the parts and uses the similarity between those vectors to assess similarity. To a certain degree the TFIDF measure also reflects the focus on objects' parts, as their names are included in the objects' full-text descriptions (see Figure 1) and as such implicitly get taken into account when computing the similarities.

We have found that the similarity measures indeed capture a sense of object similarity in an ontology. One group of measures captures the object's position within the ontology, essentially using the quality of the ontology structure as a measure for similarity. The second group practically ignores the ontology but relies on the object's structure as a measure for similarity. Only the TFIDF measure indirectly combines both approaches by implicitly including some ontology structure: some attributes and their values are inherited down the hierarchy, are passed from classes to their subclasses, and, hence, included in the textual description of both. This finding shows us that the similarity measures follow an intuitively comprehensible rationale. We do, however, still have to show that they can be useful in practical applications. We will address this issue in the next subsection in the context of ontology-aided object retrieval, a central task in applications such as the semantic web or best practice repositories.

Similarity Measure Evaluation in Ontology-Aided Object Retrieval

To answer the question about the usefulness of similarity measures, we decided to put them to use in a practical application scenario. Susan's—our consultant's—task of finding a novel sales process provides the frame for that evaluation. In particular, what if Susan were to be looking for a sales process in which customers are contacted electronically over the Internet? Such processes are very common. Online retailers such as Amazon use them every day. Susan, however, needs to find a sales process that avoids creating an untrustworthy impression on potential and current customers through unsolicited messages by considering opt-out lists. Recall that we treat complex objects and not full-text documents. Hence, Susan searches a database of processes encoded as structured data. Given this scenario, we picked a suitable retrieval technology and evaluated how similarity measures could be used to improve both its recall and precision on queries that Susan might pose.

[5] Even though the vector space assessment mechanism correlated significantly with both groups, we decided to include it in the second group, as its correlation with that group is higher.

The Base Retrieval Technology: Process Query Approach

Current technology for process retrieval has been extensively reviewed by Klein and Bernstein (Bernstein and Klein 2002; Klein and Bernstein 2004). *Keyword-based* retrieval methods rely on technology traditionally used by Web search engines (Baeza-Yates and Ribeiro-Neto 1999; Brin and Page 1998; Salton and McGill 1983) to search ontologies. These approaches typically achieve fairly high recall but low precision, as keywords are a poor way to capture the semantics of a query or item. *Table-based* approaches (Devanbu et al. 1991; Karp 2000; Fernandez-Chamizo et al. 1995; Fugini and Faustle 1993; Henninger 1995) use attribute value pairs describing the properties of an item. Both items and queries are described as tables: matches represent items whose property values match those in the query. The usefulness of these approaches is limited, however, because of the impoverished range of information typically captured by table-based service models. In *deductive retrieval* (Chen et al. 1993; Kuokka and Harada 1996; Meggendorfer and Manhart 1991), service properties are expressed formally using logic. Retrieval then consists of finding the items that can be *proven* to achieve the functionality described in the query. This approach, however, faces two very serious practical difficulties: it can be prohibitively difficult to model the semantics of non-trivial processes using formal logic and the proof process implicit in this kind of search can have a high computational complexity, making it extremely slow (Meggendorfer and Manhart 1991). Last, *structure-based* approaches, which Klein and Bernstein propose themselves (Bernstein and Klein 2002; Klein and Bernstein 2004), allow specifying queries using an ontology. They find that this last approach has excellent recall characteristics without the precision penalty, which is indigenous to keyword-based approaches. Furthermore, it is shown to have acceptable computational complexity and encompass the properties of table-based queries. Given those findings, we chose to use this last approach as a basis for our investigation.

The proposition of Klein and Bernstein is to employ a process query language (PQL) to pose queries for evaluation against a process ontology. The process query language essentially allows the composition of process fragments that result in a query-by-example style specification of the sought-after processes. In Susan's case we could assume that she would compose two queries shown as PQL and as keyword equivalents (see Table 2).

Table 2. Full-Text, Keyword-Based, and PQL Queries Used for Evaluation
(Variables in PQL queries are denoted with a question mark "?")

Query 1	
Find all sales processes that inform their customers over the Internet *Keywords:* "sell inform internet"	*PQL:* (ATTRIBUTE "Name" OF ?process INCLUDES "sell") ∧ (RELATION ?process HAS-PART ?subtask *) ∧ (ATTRIBUTE "Name" OF ?subtask INCLUDES "inform customer") ∧ (ATTRIBUTE "Mechanism" OF ?subtask INCLUDES "internet")
Query 2	
Find all sales processes that inform their customers over the internet and allow them to avoid unwanted solicitations using opt-out lists *Keywords:* "sell inform customer internet unwanted opt-out"	*PQL:* (ATTRIBUTE "Name" OF ?process INCLUDES "sell") ∧ (RELATION ?process HAS-PART ?subtask *) ∧ (ATTRIBUTE "Name" OF ?subtask INCLUDES "inform customer") ∧ (ATTRIBUTE "Mechanism" OF ?subtask INCLUDES "internet") ∧ (RELATION ?subtask HAS-EXCEPTION ?exception) ∧ (ATTRIBUTE "Name" OF ?exception INCLUDES "unwanted") ∧ (RELATION ?exception IS-AVOIDED-BY ?handler) ∧ (ATTRIBUTE "Name" OF ?handler INCLUDES "opt-out")

As we can see, query 1 is looking for sales processes that inform their customers over the Internet and query 2 adds an additional constraint of the "opt-out" list. In order to evaluate the performance of the query mechanism, we needed a coding of all 5,000 entries in the Process Handbook database as either answers (true positives) or not (true negatives) with respect to those two queries.

When running both the PQL and the keyword-based query, we find the following results. For query 1, PQL finds 17 returns of which two are correct answers and one correct answer is missing, a TFIDF-based query engine finds the true answers at the 8th, 86th, and 233rd position resulting in poorer precision and only eventually better recall. When running query 2, PQL finds two returns (both of which are correct answers) resulting in a perfect precision but a limited recall of two-thirds. The TFIDF-based algorithm finds the returns at the 3rd, 16th, and 101st position, again resulting in poorer precision and only eventually better recall. These findings essentially reflect those of Klein and Bernstein, the main difference being a new addition to the ontology.

Improving Recall: Similarity-Based Result Set Extension

We first address how we could use the similarity measures to improve the recall performance of PQL without degenerating its precision too much. Assuming that the PQL query would find at least some true positive returns, one could argue that the rest of the correct answers should be similar to the returns already found. Consequently, we decided to rank the remaining processes in the ontology in decreasing similarity to the ones found by the PQL query. In other words, we took the set of answers $\mathbb{A}$ returned by the PQL query and gave each of those a score of 1. We then calculated the score of the remaining entries in the ontology $\mathbb{R}$ by using the maximum (or average) similarity to all elements of $\mathbb{A}$. Thus the rank of $r \in \mathbb{R}$ is determined by either the maximum or average of $\{ \forall a \in \mathbb{A}: \text{similarity}(a,r) \}$. The recall of the resulting ranked returns for query 1 can be found in Figure 3 together with the original PQL and keyword-based retrieval approaches. Query 2 has analogous results. Note that we couldn't use the information-theoretic measure for these experiments, as it requires each element compared to have descendants, which is not given for leaves of the ontology—a property shared among all true (positive) answers.

As is clearly visible, the traditional keyword-based TFIDF approach provides the poorest recall performance up until the 86th returned object, where it attains the same level as PQL, which only returns 17 objects. The recall enhancing PQL methods attain the same recall as PQL until the 17th returned object. There, the recall enhancements based on the maximum similarity with either the (weighted or unweighted) edit and the vector distance immediately climb to 100 percent recall. In other words, the first element outside the PQL result set is found to be the missing true positive. As a consequence, we can argue that the above mentioned assumption of the similarity between the objects in the answer set $\mathbb{A}$ and the third true positive seems to be true. As the figure also shows the other similarity-enhanced methods follow to the 100 percent recall between the 20th and 59th returned object (20th–ontology distance max, 25th–ontology distance avg, 37th–vector distance avg, 58th–edit distance avg, 59th–weighted edit distance avg), whereby the methods based on average similarity are clearly outperformed by those based on maximum similarity. Summarizing, we can state that our approach for enhancing recall has clearly worked on the example queries and seems to merit further investigation.

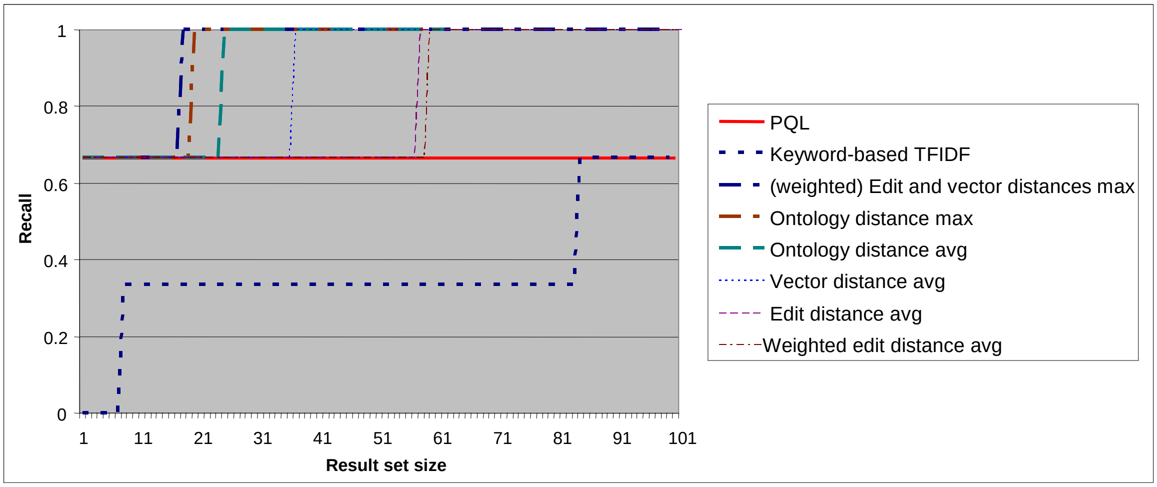


Figure 3. Recall of Traditional Keyword-Based, PQL, and Similarity Enhanced PQL Approaches for Query 1

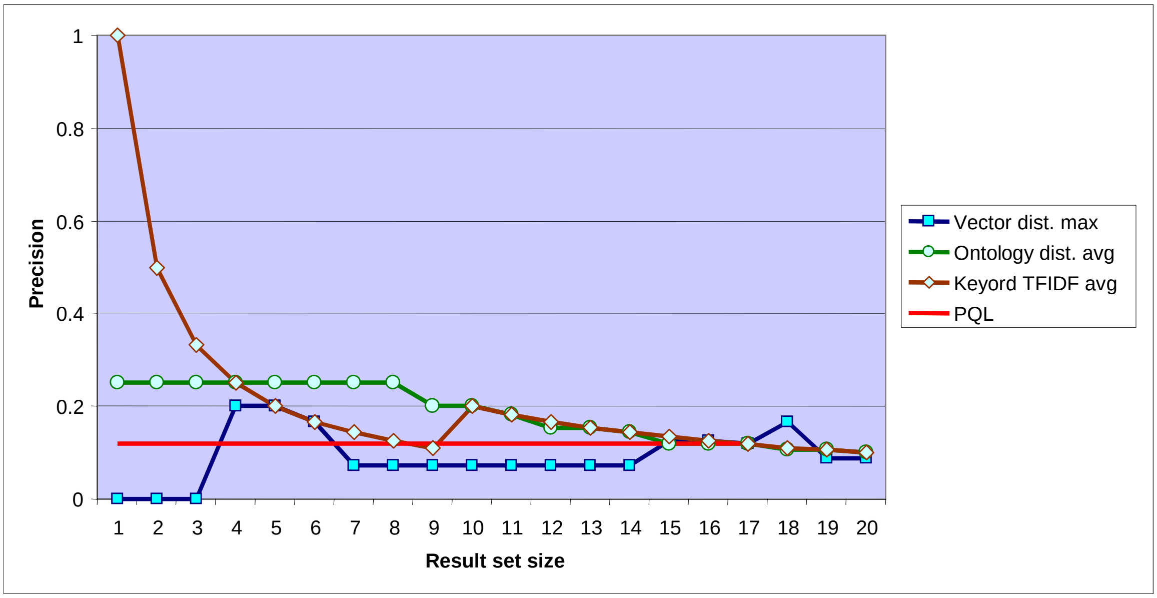


Figure 4. Precision Analysis of Traditional Keyword-Based, PQL, and Enhanced PQL Approaches for Query 1
(Non-improving methods are omitted to improve chart clarity)

Improving Precision: Similarity-Based Result Set Ordering

Given our success in enhancing recall, we devised methods for using object similarity measures for reordering the PQL result set with the goal of enhancing retrieval precision. Again we based our approach on the assumption that the result set returned by the PQL method contained some information about the nature of the true answers to Susan's question. As a consequence, we took the elements in the answer set $\mathbb{A}$ and ordered it by the maximum (or average) similarity to the other elements in $\mathbb{A}$. Thus the rank of $a \in \mathbb{A}$ is defined as maximum (or average) of $\{\forall b \in \mathbb{A}\text{: } a \neq b : \text{similarity}(b,a)\}$. As an additional measure we decided to order the elements of $\mathbb{A}$ by their keyword-based TFIDF score.

The results of this precision analysis for query 1 can be found in Figure 4, which clearly shows that precision can be substantially improved with the suggested approach. As a baseline, we have to take the pure PQL results, which are represented as the fat red line at a precision of 0.118. Note also, that PQL returns 17 answers to the query, at which point almost all methods meet as they base on the original PQL query. It is clearly visible, however, that some methods, such as the average TFIDF (shown as brown with diamonds), ontology (green with circles), or vector (blue with squares) similarity enhanced PQL methods, successfully order the returns to increase precision. Summarizing, we find that similarity measures, again, showed their capability to improve retrieval performance—in this case, precision—of complex objects in ontologies. Given the already perfect precision of query 2 we omit its discussion here.

Combination: An Improved Similarity-Aided Object-Retrieval Approach

So far we have shown that similarity measures between complex objects can be successfully used to increase the precision as well as the recall of a successful ontology-based object retrieval method. Obviously, those two approaches can be combined to result in an overall well-performing retrieval method, in which the elements in the answer set $\mathbb{A}$ are ranked according to their average ontology distance or TFIDF similarity and the ones in $\mathbb{R}$ are ranked (for example) by their edit distance similarity. The result of this combination should be an overall well-performing retrieval mechanism. The remaining question is how to choose the best overall performing similarity measures for the ranking. In the information retrieval literature, combinations of precision and recall are typically evaluated using the *F*-measure—the harmonic mean of precision and recall—or the *E*-measure[6] (see Baeza-Yates and Ribeiro-Neto 1999, p. 82). Both measures assume that the tradeoff between precision and recall made by the users of the system is known at the time of the analysis; the *F*-measure weighing them equally, the *E*-measure using a parameter to determine

[6] Calculated as $1 - (1 + b^2) / (b^2/\text{recall} + 1/\text{precision})$, where b provides a weight of precision versus recall.

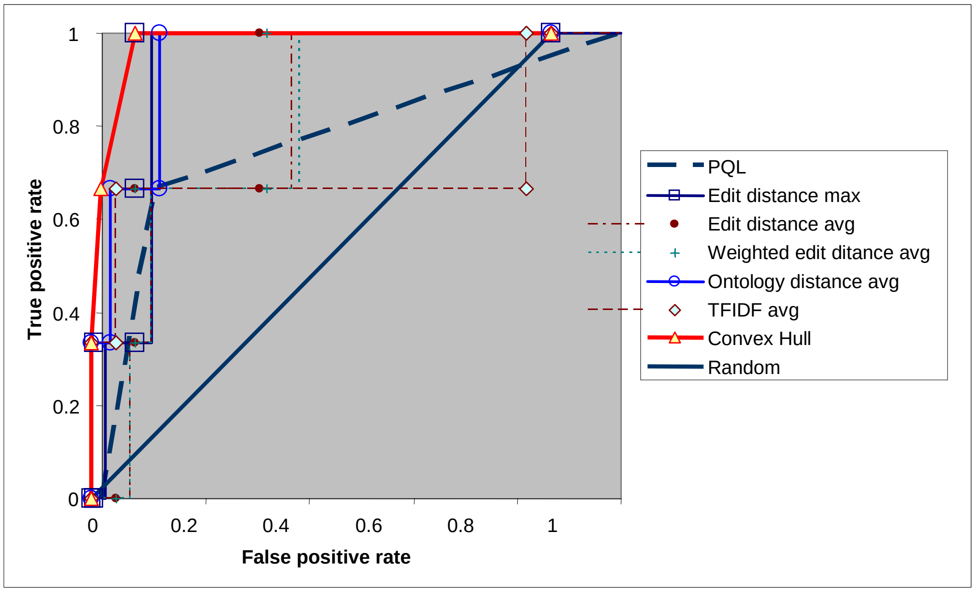


Figure 5. Receiver Operating Characteristics of PQL and Some Similarity Enhanced PQL Methods for Query 1

the weight of the tradeoff. As we do not precisely know the weight given to either precision or recall, we draw on the receiver operating characteristic (ROC) analysis, which has recently gained usage in the machine learning literature (Provost and Fawcett 2001). ROC curves graph the false positive rate versus the true positive rate of a classification or retrieval operation. As such, the curves, which are limited to the space between [0,0] and [1,1], allow the comparison of different retrieval mechanisms regardless of the cost of false positives or false negatives and the distribution of the true positives in the underlying dataset.

Consider, the ROC curves for our retrieval methods in Figure 5. As we can see, all of the methods' curves are in the upper left half of the graph. This is a good sign, as the perfect method would show as a curve from the origin [0,0] through [0,1] to [1,1], while the random method would lie on the diagonal providing a baseline. The closer to the ideal curve, the better the method. Hence, the method's quality is oftentimes measured as the area under the ROC curve. As we can see, some of our methods perform very well. The maximum edit distance enhanced PQL method (highlighted with empty squares), for example, follows the y-axis very closely until about 0.3, then steps to [0.1, 0.3] and [0.1, 1] before it reaches [1,1]. Some other methods, such as the average ontology distance (empty circles), clearly outperform it between x = 0 … 0.1, but fall short after that region. This shows how the ROC curves capture the two methods' performance with regard to precision and recall, where the maximum edit distance method outperformed the average ontology distance in the recall evaluation and vice versa: in ROC curves, the relationship between the cost false positives and false negatives determines the incline of a line. The method whose tangent to that line lies closest to [0,1] is the optimal under the given cost relationship. In our example, the incline, thus, essentially captures the weight given to precision and/or recall. As Provost and Fawcett (2001) show, an overall optimally performing method can, therefore, be constructed by combining the methods that together establish the convex hull (shown fat with triangles). Summarizing, we have seen that the similarity measures for complex objects as introduced above can indeed be used to improve retrieval performance of objects as well as provide the basis for constructing a combined, overall well-performing method.

Limitations

The above evaluation illustrates how similarity measures can be used to improve both precision and recall of complex objects in an ontology. As such, it fulfills the goal that we set ourselves to show the usefulness of the similarity measures both through a comparison within a realistic ontology and in the context of an application. Nevertheless, there are limitations to the evaluation. First, and concerning both evaluations shown, the limitation on one ontology might limit the generalizability of the results. Budanitsky and Hirst (2001), for example, state that the differences in retrieval performance found in various NLP publications might stem from the different versions of WordNet used as the underlying ontology. As such, our evaluation needs to be extended to include other ontologies before a general statement can be made. But even in its current form, our findings on the different notions of similarity and methods for extending current object retrieval methodologies with object similarity measures are highly likely to generalize to other domains, as some of the similarity measures have been adopted from other domains where they have been used successfully in the context of various ontologies. Second, and concerning the retrieval evaluation (only), our limitation on two queries hampers validity to queries in general. While we will have to address this concern with broader evaluations in the future, we believe that findings from other domains showing the usefulness of similarity measures for retrieval of text documents or for clustering feature vectors should generalize to the use of similarity measures of complex objects supporting our illustration in an object or process retrieval application. Third, our repertoire of similarity algorithms is far from complete. It presents but a first version of a framework which we intend to extend significantly. Notable exceptions, for example, are graph or tree-based similarity algorithms (Guha and McCool 2003; Jonyer et al. 2001; Melnik et al. 2002; Palopoli et al. 2003; Roddick et al. 2003; Wang et al. 1999). Last, we did not evaluate the computational performance of the algorithms—an issue we need to address in future work.

Related and Future Work

Most closely related, Rodriguez and Egenhofer (2003), similar to Fridman Noy and Musen (1999), present an approach to computing semantic similarity that accounts for differences in the levels of explicitness and formalization of different ontologies. They combine three similarity measures in order to detect similar classes *across ontologies* and find that the combination of word and semantic-neighborhood matching obtains best recall and precision rates, whereas combining word matching with feature matching results in increasing precision, but decreasing recall. Each measure on its own is insufficient for cross-ontology evaluations. This is similar to our finding that the best retrieval performance is likely to be found by combining different measures. Given their focus on comparisons across ontologies, however, it is difficult to compare our results.

Di Noia et al. (2003) compare a human-based ranking of 12 items with the returns of a description logic-based retrieval engine, which attains imprecise matching by relaxing query constraints. This is similar to using an ontologized edit distance for ranking retrieved objects. They find the automated rankings show good correspondence to the average human subject's assessment and refer to ongoing large-scale experiments for further details. Their work differs from ours in that they do not compare their ranking method with any other similarity measures, preventing a comparison with the methods we presented. Nonetheless, in future work we also intend to evaluate retrieval performance with human subjects.

Ouzzani and Bouguettaya's (2004) propose and implement a generic approach for optimally querying Web services using their input/output parameters, whose sole use for retrieval has been shown to be problematic (Klein and Bernstein 2004). They don't report any evaluation of their approach. Nevertheless, *their study shows the big impact good similarity measures might have on practical applications*.

Focusing on a real-world application in the domain of bioinformatics, Lord et al. (2003) found that sequence similarity of proteins correlated well with Resnik's information-content-based similarity operating on protein annotations. Using the semantic measure, they generated a ranked list of semantically similar proteins to enhance recall. Their main problem was that many of the results had identical similarity values and can, therefore, not be ranked appropriately. We believe that the use of a more fine-grained similarity algorithm could help to address this issue (e.g., by sorting similar ranked items as we did in the recall experiment).

The NLP literature provides a large group of related work *focusing on word rather than object similarity*. Motivated by Resnik's study, a number of papers describe improvements to his information-theoretic measure. Wu and Palmer (1994) focus on the semantic representation of verbs in computer systems and find those measures well applicable in the field of machine translation. Jiang and Conrath (1997) propose a combined edge-counting and node-based method that outperforms either of the pure approaches. This hints at the usefulness of different or combined similarity algorithms for different notions of similarity, just as we found it when constructing the combined precision *and* recall enhancing method. Furthermore, Budanitsky and Hirst (2001) evaluate five measures of semantic relatedness in a real-world malapropism detection and spelling correction system. They show that malapropism detection proves to be an effective approach for evaluating similarity measures and that there are considerable differences in the performance of the measures. These studies, thus, support our finding that there are different notions of similarity and, therefore, that different semantic similarity measures aren't equally suitable for all applications.

A number of studies explore similarity measures in general. Lin (1998) explores an information-theoretic measure of similarity that relies on a probabilistic model of the application domain, which makes it problematic for smaller ontologies. Roddick et al. (2003) propose a graph-based approach that determines the semantic distance between objects through a traversal distance, weighted arcs, and transition costs. They show that their approach can be simplified to straightforward enhancements to standards such as SQL.

Already in the late 1980s concept-based retrieval systems were proposed to enhance the precision and recall of text documents. In Giger's (1988) study, for example, users enter queries as keywords and a concept space (derived from a corpus) together with a cosine/entropy-based similarity function are used to improve retrieval performance. The focus of that stream of research on documents rather than complex objects differentiates it from our study.

Cohen's (2000) WHIRL system uses a TFIDF-based similarity operator to enhance a relational database system allowing similarity-based joins. WHIRL illustrates the power of similarity functions in a context of structured objects (in his case, database relations). It differs from our approach in that WHIRL compares attributes of database records for joins, whereas we compare the similarities of whole complex objects (e.g., database records).

Summarizing, we can say that the study of similarity measures has been found to be an important subject of research in many domains of computer science and information systems. Its impact in some of the domains such as NLP and information retrieval cannot be overstated. Some of the more recent studies also show that similarity measures for objects are likely to have a similar impact on practical application domains such as the semantic web, Web-service discovery, object-retrieval, or the use of best practice repositories. Much work remains to be done. Our catalogue of similarity measures should be extended with other candidates mentioned in this section and beyond. Furthermore, we need to address the limitations of our evaluation by extending it to more queries, evaluating them in differing ontologies, and comparing their assessments to human subjects. Last but not least, we intend to compare the usefulness of the similarity measures in other applications such as clustering, ontology construction, and others.

Conclusions

In this paper, we argued that similarity measures between complex objects in ontologies, a central component of techniques such a clustering, data-mining, semantic sense disambiguation, ontology translation, and simple object comparison, deserve more attention given their foundational value for business-relevant computing tasks like finding processes in best practice repositories, finding a suitable service provider or outsourcing partner for agile process enactment, dynamic customer segmentation, and systems integration. We assembled a catalogue of five algorithms (one of which was presented in two versions) and compared them to each other within a large, real-world best practice ontology as well as evaluated them in a realistic business process retrieval scenario. We found that different similarity algorithms reflect different notions of similarity, which arise from the comparison of complex objects rather than documents (as bags of words) or feature vectors. This intriguing finding hints at the importance of people's cognition of object similarity, which, like the algorithms' similarity assessments, might have several aspects and, as such, warrants further explorations. We also showed how a combination of similarity measures can be used to improve both precision and recall of an existing object-retrieval approach providing a notion of practical applicability of the similarity measures found and ultimately helping practitioners such as Susan in a practically relevant task. This study provides a first investigation of similarities in ontologies. Nevertheless, the task of understanding similarity in ontologies is far from over. To that end, both technical work on better, feature-combining, and ontology-adapting similarity assessment algorithms as well as behavioral studies exploring people's understanding of similarity and their use of similarity-based features are needed.

Acknowledgments

The authors would like to thank the MIT Process Handbook project for making available the data on which the evaluation is based, and the anonymous reviewers for their helpful comments. This work was partially supported by the Swiss National Science Foundation grant 200021-100149/1.

References

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (Eds.). *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, Cambridge, UK, 2003.

Baeza-Yates, R., and Ribeiro-Neto, B. d. A. *Modern Information Retrieval*, Addison-Wesley, Reading, MA, 1999.

Bernstein, A., and Klein, M. "Discovering Services: Towards High-Precision Service Retrieval," in *Proceedings of the Web Services, E-Business and Semantic Web Workshop at the Fourteenth International Conference on Advanced Information Systems Engineering* (CAiSE-2002), C. Bussler, R. Hull, S. A. McIlraith, M. E. Orlowska, B. Pernici, and J. Yang (Eds.), Springer Lecture Notes in Computer Science Vol. 2512, New York, 2002, pp. 260-275.

Brin, S., and Page, L. "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems* (30:1-7), 1998, pp. 107-117.

Budanitsky, A., and Hirst, G. "Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures," paper presented at the second meeting of the North American Chapter of the Association for Computational Linguistics (NAACL2001), Pittsburgh, PA, 2001.

Chen, P., Hennicker, R., and Jarke, M. "On the Retrieval of Reusable Software Components," in *Proceedings Advances in Software Reuse. Selected Papers from the Second International Workshop on Software Reusability*, 1993, pp. 99-108.

Cohen, W. W. "Data Integration Using Similarity Joins and a Word-Based Information Representation Language," *ACM Transactions on Database Systems* (18:3), 2000, pp. 288-321.

Devanbu, P., Brachman, R. J., Selfridge, P. G., and Ballard, B. W. "LaSSIE: A Knowledge-Based Software Information System," *Communications of the ACM* (34:5), 1991, pp. 34-49.

Di Noia, T., Di Sciascio, E., Donini, F. M., and Mongiello, M. "A System for Principled Matchmaking in an Electronic Marketplace," paper presented at the WWW2003 Conference, Budapest, Hungary, 2003.

Dzeroski, S., and Lavrac, N. *Relational Data Mining,* Springer, Berlin, 2001.

Fernandez-Chamizo, C., Gonzalez-Calero, P. A., Hernandez-Yanez, L., and Urech-Baque, A. "Case-Based Retrieval of Software Components," *Expert Systems with Applications* (9:3), 1995, pp. 397-421.

Fridman Noy, N., and Musen, M. A. "SMART: Automated Support for Ontology Merging and Alignment," paper presented at the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada, 1999.

Fugini, M. G., and Faustle, S. "Retrieval of Reusable Components in a Development Information System," in *Proceedings of the Second International Workshop on Software Reusability*, IEEE Press, Los Alamitos, CA, 1993, pp. 89 - 98.

Giger, H. P. "Concept Based Retrieval in Classical IR Systems," in *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Grenoble, France, 1988, pp. 275-289.

Guha, R., and McCool, R. "TAP: A Semantic Web Platform," *Computer Networks* (42:5), 2003, pp. 557-577.

Henninger, S. "Information Access Tools for Software Reuse," *Journal of Systems & Software* (30:3), 1995, pp. 231-247.

Jarmasz, M., and Szpakowicz, S. "Roget's Thesaurus and Semantic Similarity," TR-2003-01, Technical Reports, University of Ottawa School of Information and Technology, Ottawa, ON, Canada, 2001.

Jiang, J. J., and Conrath, D. W. "Semantic Similairty Based on Corpus Statistics and Lexical Taxonomy," paper presented at the International Conference on Research on Computational Linguistics (ROCLING X), Taiwan, 1997.

Jonyer, I., Holder, L. B., and Cook, D. J. "Graph-Based Hierarchical Conceptual Clustering," *International Journal of Artifical Intelligence Tools* (10:1-2), March 2001, pp. 107-135.

Karp, A. "E-speak E-xplained," HPL-2000-101, Technical Report, Hewlett-Packard Labs, Palo Alto, CA, 2000.

Klein, M., and Bernstein, A. "Towards High-Precision Service Retrieval," *IEEE Internet Computing* (8:1), January 2004, pp. 30-36.

Kuokka, D. R., and Harada, L. T. "Issues and Extensions for Information Matchmaking Protocols," *International Journal of Cooperative Information Systems* (5), 1996, pp. 2-3.

Lee, J. H., Kim, M. H., and Lee, Y. J. "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies," *Journal of Documentation* (49:2), 1993, pp. 188-207.

Levenshtein, V. I. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady* (10), 1966, pp. 707-710.

Lin, D. "An Information-Theoretic Definition of Similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98),* Madison, WI, 1998.

Lord, P. W., Stevens, R. D., Brass, A., and Goble, C. A. "Investigating Semantic Similarity Measures Across the Gene Ontology: The Relationship between Sequence and Annotation," *Bioinformatics* (19:10), 2003, pp. 1275-1283.

Malone, T. W., Crowston, K., and Herman, G. A. (Eds.). *Organizing Business Knowledge: The MIT Process Handbook*, MIT Press, Cambridge, MA, 2003.

Malone, T. W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., and O'Donnell, E. "Tools for Inventing Organizations: Toward a Handbook of Organizational Processes," *Management Science* (45:3), 1999, pp. 425-443.

McCallum, A. K. "Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering.," unpublished manuscript, 1996 (available online at **http://www.cs.cmu.edu/~mccallum/bow**; accessed September 20, 2004).

Meggendorfer, S., and Manhart, P. "A Knowledge and Deduction Based Software Retrieval Tool," in *Proceedings of the 6th Annual Knowledge-Based Software Engineering Conference*, IEEE Press, Los Alamitos, CA, 1991, pp. 127-133.

Melnik, S., Garcia-Molina, H., and Rahm, E. "Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching," paper presented at the 18th International Conference on Data Engineering (ICDE), San Jose, CA, 2002.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. "Introduction to WordNet: An On-line Lexical Database," Technical Report, Cognitive Science Laboratory, Princeton University, Princeton, 1993 .

Miller, G. A., and Charles, W. G. "Contextual Correlates of Semantic Similarity," *Language and Cognitive Processes* (6:1), 1991, pp. 1-28.

Mitchell, T. M. *Machine Learning*, McGraw-Hill, New York, 1997.

Ouzzani, M., and Bouguettaya, A. "Efficient Access to Web Services," *IEEE Xplore: Internet Computing* (8:2), March-April 2004, pp. 34-44.

Palopoli, L., Saccà, D., Terracina, G., and Ursino, D. "Uniform Techniques for Deriving Similarities of Objects and Subschemes in Heterogeneous Databases," *IEEE Transactions on Knowledge and Data Engineering* (15:2), 2003, pp. 271-294.

Provost, F., and Fawcett, T. "Robust Classification for Imprecise Environments," *Machine Learning* (42), 2001, pp. 203-231.

Rada, R., Mili, H., Bicknell, E., and Bletner, M. "Development and Application of a Metric on Semantic Nets," *IEEE Transactions on Systems, Man, and Cybernetics* (19:1), 1989, pp. 17-30.

Resnik, P. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," paper presented at the 14th International Joint Conference on Artificial Intelligence, Montreal, 1995, pp. 448-453.

Resnik, P. "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language," *Journal of Artificial Intelligence Research* (11), 1999, pp. 95-130.

Roddick, J. F., Hornsby, K., and de Vries, D. "A Unifying Semantic Distance Model for Determining the Similarity of Attribute Values," in *Proceedings of the 26th Australasian Computer Science Conference on Research and Practice in Information Technology*, Adelaide, Australia, 2003, pp. 111-118.

Rodriguez, M. A., and Egenhofer, M. J. "Determining Semantic Similarity Among Entity Classes from Different Ontologies," *IEEE Transactions on Knowledge and Data Engineering* (15:2), February 2003, pp. 442-456.

Sachs, L. *Angewandte Statistik* (10th ed.), Springer, Berlin, 2002, p. 889.

Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval,* McGraw-Hill, New York, 1983.

Wang, J. T. L., Zhang, K., and Chang, C.-Y. "Identifying Approximately Common Substructures in Trees Based on a Restricted Edit Distance," *Information Sciences* (121:3-4), 1999, pp. 367-386.

Wu, Z., and Palmer, M. "Verb Semantics and Lexical Selection," in *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, New Mexico, 1994, pp. 133-138.