A Method for Determining Ontology-Based Semantic Relevance

Tuukka Ruotsalo and Eero Hyvönen

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
firstname.lastname@tkk.fi
http://www.seco.tkk.fi/

Abstract. The semantic web is based on ontologies and metadata that indexes resources using ontologies. This indexing is called annotation. Ontology based information retrieval is an operation that matches the relevance of an annotation or a user generated query against an ontology-based knowledge-base. Typically systems utilising ontology-based knowledge-bases are semantic portals that provide search facilities over the annotations. Handling large answer sets require effective methods to rank the search results based on relevance to the query or annotation. A method for determining such relevance is a pre-requisite for effective ontology-based information retrieval. This paper presents a method for determining relevance between two annotations. The method considers essential features of domain ontologies and RDF(S) languages to support determining this relevance. As a novel use case, the method was used to implement a knowledge-based recommendation system. A user study showing promising results was conducted.

1 Introduction

The semantic web [4] promotes the use of explicit background knowledge (metadata) to manage diverse resources. Metadata has a defined meaning in terms of a domain ontology that provides a shared conceptualisation of the domain of discourse [9]. Resources are indexed using metadata schemas and values from domain ontologies. Resources indexed with ontological values are called annotations. While the research of the logical structure of the ontologies and metadata schemas has gained much popularity in the past years, the methods for information retrieval have mainly concentrated on strict boolean querying of a knowledge-base rather than assessing relevance for the annotations in the knowledge-base. Good examples can be found in a field of semantic portals [14], that provide search facilities to access the data. Many of the portals so far have utilised search facilities that are based on Boolean queries or facet-based search [2, 16]. To enable effective information retrieval, methods for ranking and clustering the search results are a necessity.

The relevance determination problem has an important background in text retrieval where document-term matrices are used to calculate similarity of the documents [5]. Good results have been achieved with *tf-idf* weighting of the feature vectors [19, 3].

The majority of current research in ontology-based information retrieval has focused on crisp logic with intelligent user interfaces to formulate the query [16]. These have been further developed to support fuzzy logic [10]. Determining structural similarity has been investigated in SimRank [13], SimFusion [20], AKTiveRank [1] and Swoogle [8]. SimRank measures similarity of structural contexts, but concentrates only on graph theoretical model instead of feature vectors. SimFusion considers object features, but does not bind the features to ontologies. The Swoogle search engine uses *term rank* and *onto rank* algorithms to provide the relevance to predict correct ontology and instances for terms and concepts. However, Swoogle concentrates on matching classes and terms to ontologies, but does not consider the mutual relevance of annotations. AKTiveRank uses semantic similarity and the alignment of the terms in separate ontologies as a ranking principle.

In this paper we present a method that calculates the mutual relevance of annotations. Unlike SimRank, SimFusion, Swoogle and AKTiveRank we concentrate on the relevance of the annotations based on the underlying domain ontology. We extend the *tf-idf* [19, 3] method by considering essential features of the domain ontologies and RDF(S) languages. The method can be used for numerous applications such as knowledge-based recommendation [15, 7], information retrieval and clustering [3]. As a novel use case we present a recommendation system implemented with a real-life dataset. Finally, we show initial empirical results from a user test that support the method.

2 Knowledge Representation on the Semantic Web

2.1 Representation of Annotations

The Semantic web contains metadata about resources. This metadata is called annotations. Annotations are formulated with a RDF(S) [6] language, where each statement about the resource is given as an *annotation triple*. A set of annotation triples describe a resource x.

An annotation is a set of triples $E = \{< subject, predicate, object >\}$, where at least one $subject \equiv x$ (i.e. it is connected to the resource that is being annotated). Triples that have a resource as the object value are called *ontological elements*; triples with literal values are called *literal elements*. In this paper, we are concerned with ontological relevance and therefore only ontological elements are considered.

In addition to the triples, RDF Schema language (RDFS) defines a schema for RDF. RDFS separates classes and instances. For example a resource *GeorgeWBush* could be defined to be an instance of the classes *Person* and *President*. In RDFS classes can be defined as subsumption hierarchies. For example class *President* could be defined to be a *subClassOf* a class *PoliticalRole*.

We next present requirements for a method by which the ontology-based semantic relevance between resources can be determined.

2.2 Requirements for Annotation Relevance Calculation

To fully support the data model behind RDF(S), the following criteria must be taken into account by the method determining the relevance:

1. **Classes and instances**. A typical approach in knowledge representation is to separate classes and instances. For example, when annotating a web page with the resource *'GeorgeWBush'*, the particular instance of a class, say *'Politician'*, is referred to. Any other annotation stating something about the same resource *'GeorgeWBush'* would also refer to this particular instance. The instance sharing approach leads to undisputed benefits because the resources have a unique identifier. However, if the instance is commonly referred in the knowledge-base it may over-dominate traditional retrieval methods.
2. **Subsumption**. Concepts in the domain ontologies are typically ordered in subsumption hierarchies. For example, if *'GeorgeWBush'* is an instance of the class *'Politician'* this could be subsumed by the class *'Person'*. It is clear that the fact that it is implicitly known that *'GeorgeWBush'* is also related to class *'Person'* has to be taken into a consideration by information retrieval methods, but intuitively with less relevance than the class *'Politician'*, since persons may also be non-political.
3. **Part-of relations**. In addition to subsumption, many domain ontologies introduce relations to support the theory of parts and wholes (part-of). For example, if a resource is annotated with the instance *'New York City'*, it may be relevant in the scope of *'New York State'* due to the part-of-relation between the resources, although the notion of the state does not subsume the notion of the city. Part-of relations are useful in information retrieval, but require separate handling from subsumption relations [17].

Next we present the method for determining ontology-based annotation relevance where these requirements are taken into consideration.

3 A Method for Determining Semantic Relevance of Annotations

Consider resources x and y. The ontological relevance r of a resource y, when a resource x is given, is defined by the quadripartite relation S

$$S \subset \{< x, y, r, e > | x \in C, y \in C, r = ar(ann(x), ann(y)) \in [0, 1], e \text{ is a literal}\},$$

where C is the set of resources, ar is a real valued function *annotation relevance* expressing how relevant y is given x, ann is a function returning annotation triples for a resource, and e is a literal explanation of why y is relevant given x. A tuple $< x, y, r, e > \in S$ intuitively means that "x is related to item y by relevance r because of e". For example:

<GeorgeWBush, WhiteHouse, 0.8, "George W. Bush workingIn Whitehouse">

The relevance relation can be used in semantic recommending: it provides the set of explained recommendations for each content item x. In addition, the relevance relation could be used for clustering or as a search base of its own if the end-user is interested in finding relations between resources instead of resources themselves.

Below, we first present a method for computing the annotation relevance $r = ar(ann(x), ann(y))$ for resources x and y, and then discuss how to provide the explanation e.

A widely used method for determining the relevance of a document with respect to a keyword k is *tf-idf* [3]. Here the relevance r_k of a document d with respect to k is the product of term frequency (*tf*) and inverse document frequency (*idf*) $r_k = tf \times idf$. The term frequency $tf = n_k/n_d$ is the number of occurrences of k in d divided by the number n_d of terms in d. The inverse document frequency is $idf(d) = log\frac{N}{N_k}$ where N is the number of documents and N_k is the number of documents in which k appears. Intuitively, *tf-idf* determines relevance based on two components: *tf* indicates how relevant k is w.r.t. d and *idf* lessens the relevance for terms that are commonly used in the whole document set.

Our case is different from the classical text document retrieval in the following ways. First, the document set is a set of ontological annotations. The *tf* component cannot be based on term frequency as in *tf-idf*. Second, we will not search for relevant documents with respect to a key word, but try to find semantically related ontological annotations. To account for these differences, we devised *idf*-like measures *inverse class factor*, *inverse instance factor*, and *inverse triple factor* that account for the global usage of classes, individuals and triples in the annotations.

Definition 1 (inverse class factor). *The inverse class factor $icf(c)$ for a class c is $icf(c) = log\frac{N}{N_c}$, where N is the total number of instances of all classes used in the annotations, and N_c is the number of instances of the class c.*

Intuitively, *icf(c)* is higher for annotation instances whose class are rarely used in annotation.

Definition 2 (inverse instance factor). *The inverse instance factor $iif(i)$ for an instance i is $iif(i) = log\frac{I}{n}$, where I is the total number of instances shared by the annotations, and n is the number of usage of the instance i.*

This measure takes into account the fact that instances can be shared by the annotations. The idea of using *iif(i)* will be to lessen relevance of content items that share same instances, when such instances are commonly used.

In order to define the inverse triple factor we first define the predicate $cmatch(x, y)$ for matching two instances and $pmactch(p, q)$ for matching two properties (rdf predicates). Let $cl(x)$ denote the class of instance x, $sp(c)$ denote the set of superclasses of class c, and $pr(p)$ denote the set of super properties of property p. Then:

$$cmatch(x, y) = \text{true, if } x = y \text{ or } cl(x) \in sp(cl(y))$$
$$pmatch(p, q) = \text{true, if } p = q \text{ or } p \in pr(q).$$

Definition 3 (inverse triple factor). *The inverse triple factor $itf(t)$ for a triple $t =< s, p, o >$ is: $itf(t) = log\frac{T}{N}$, where T is the total number of annotation triples $< s', p', o' >$, such that $cmatch(s', s)$ and $pmatch(p', p)$ and $cmatch(o', o)$ hold, and N is the total number of annotation triples.*

In addition, a measure is needed to determine the relevance between two instances based on the class membership, part-of, and an instance equivalence relations in the domain ontology:

Definition 4 (ontological instance relevance).

The ontological instance relevance of instance y given instance x is

$$oir(x,y) = \begin{cases} iif(y) \times icf(cl(y)) \text{ if } cmatch(x,y) \\ 0.70 \times iif(y) \times icf(cl(y)) \text{ if } partOf(y,x) \text{ or } partOf(cl(y),cl(x)), \\ 0 \text{ otherwise} \end{cases}$$

where $partOf(x,y)$ is true if x is a part of y. If two first cases match at the same time, the maximum value is selected.

The ontological instance relevance is given by the product of the inverse instance factor and inverse class factor. The similarity can be calculated if the instance between annotation objects is shared or the class membership of the target instance is in the transitive closure of the class membership of the source instance. In addition the target instance or target class membership can be connected to the source instance or to the source class membership with a part-of relation. We have used 0.70 as the part-of multiplier based on extensive empirical tests by Rodriquez and Egenhofer [17]. Intuitively, $oir(x,y)$ is high, i.e. y is relevant for x, when y and $sp(cl(y))$ are rarely used in annotations, and x and y are related by hyponymy, meronymy, or equivalence.

The ontological triple relevance can now be defined.

Definition 5 (ontological triple relevance). *The ontological relevance of a triple $y =< s,p,o >$, given a triple $x =< s',p',o' >$ and assuming that $pmatch(p,p')$ holds, is:*

$$otr(x,y) = oir(s,s') + oir(o,o') + itf(y).$$

Intuitively, $otr(x,y)$ is high, i.e. y is relevant for x, when the subject and object of y are relevant given the subject and object of x, respectively, and y is rarely used.

Finally the annotation relevance is the sum of the ontological triple relevances for the annotations.

Definition 6 (annotation relevance). *The annotation relevance $ar(A,B)$ of an annotation A, given an annotation B, is*

$$ar(A,B) = \frac{\sum_{a \in A, b \in B} otr(a,b)}{n_t},$$

where n_t is a number of triples in a target annotation used as a normalisation factor.

When determining the values $ar(x,y)$, the explanation literal e can be formulated based on the labels of the matching triples.

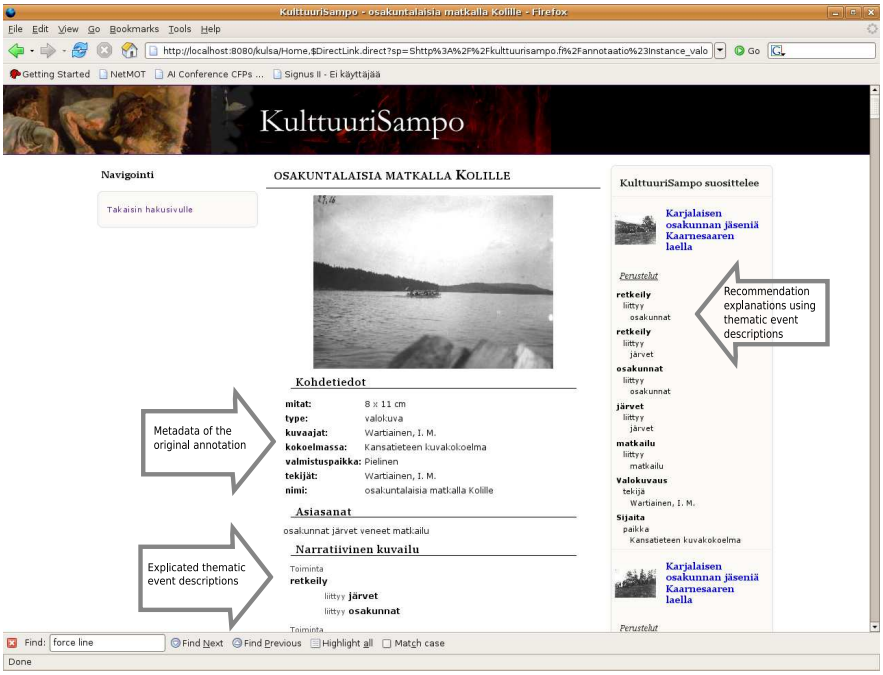

Fig. 1. CULTURESAMPO user interface showing a photograph, its metadata, and semantic recommendation links.

4 Implementation and Evaluation

The method presented above has been implemented in the recommendation system of the CULTURESAMPO prototype portal [11]. A user study was conducted to evaluate how well the method predicted the ranking of the resources compared to opinions of the users. In information retrieval systems, the users usually want to see just ten to twenty documents, and if these do not correspond to the information need of the users the search is re-adjusted [3]. This is why in practical applications, such as knowledge-based recommendation, the ranking of the documents is a crucial task.

A user study was conducted to evaluate the method. The hypothesis tested was: does the ranking performed by the annotation relevance method correspond with the end-user's opinion of the ranking. In practice this means testing whether the users liked more the recommended resources ranked higher (target documents) based on a source resource (source document) they were looking at.

The most obvious way to measure this is to calculate the correlation between the ordering of the items made by the method and by the user. Based on a preliminary user test, it turned out that the ordering of the documents was difficult for the users. However, the users indicated that it was rather easy to classify the documents into two groups: the

highly relevant and less relevant. Therefore, this simple ranking dichotomy was used in the test.

4.1 Dataset

The dataset used contained annotations of three different resources: images of museum items, images of photographs and images of paintings. These were annotated by domain experts in Finnish museums. The General Finnish Ontology (YSO) [12] was used as a domain ontology. This domain-ontology consists of more than 23.000 classes organised in subsumption and part-of hierarchies. Seven documents were randomly selected as source documents representing the source resources: two images of museum items, three images of photographs and two images of paintings. Ten target recommendations were then calculated for each source document representing the target resources, which resulted to a set of 70 target documents. The calculation was performed against a knowledge-base that contained annotations of nearly 10.000 resources of before-mentioned types.

The five top-ranked recommendation documents given by our method were considered the *higher relevance group*. The other five, the *lower relevance group*, were a sample of the lower half of the ranking based on the median relevance. To exclude highly non-relevant recommendations its was required that the source document and its target recommendation should share at least two triples.

4.2 Test Setting

Figure 1 illustrates the user interface showing a page about a photograph in a dataset. The pages were printed without the recommendations that can be seen on the right side of the figure. A card sorting experiment was conducted based on the printed pages [18]. Seven test subjects were first asked to classify the recommendations according to the seven source documents, based on the metadata and the image. After this the subjects were asked to formulate the higher relevance group and the lower relevance group for each source document. In both tests the test subjects were able to leave a target recommendation document out, if the they felt that it was not relevant given any source document.

4.3 Results

The right source document for a recommendation was found in 71 per cent (%) of the cases. Test subjects then classified the items in the higher relevance group correctly in 82% of the cases. From the documents that were classified under the wrong group, 23% were in the higher relevance group and in 77% in the lower relevance group. The share of the target recommendation documents that the test subjects were unable to classify into either the high or low relevance group, 5% were in higher relevance group, and 95% in the lower relevance group.

4.4 Conclusions of Empirical Evaluation

These results show that the method predicted the relevance very well for a first attempt: only 5% of the recommendations were left out as non-relevant. All of the recommendations left out were from the lower relevance group. Only 18% of the recommendations were classified wrongly in the higher relevance group (including the items that still belonged to the lower relevance group). Only 3% of the recommendations were classified under the wrong source item in the higher relevance group.

5 Conclusions

Previous work in knowledge-based recommendation and object relevance has focused on measuring the similarity of feature vectors [7, 15], where similarity measures are used to calculate nearest neighbours from a vector space. This approach works well in a single domain, where features can be predefined and weights assessed for features. The *tf-idf* methods have shown promising results in measuring the relevance in information retrieval from natural language documents [3].

Our work extends such measures by adopting the ontology and the annotation triples as a source for feature matching. In terms of *tf-idf*, we have extended the *idf* component to consider three essential features of ontology-based systems, namely separation of classes and instances, support for subsumption and support for part-of relations. We have implemented the method in the semantic portal CULTURESAMPO. In addition, we have conducted a user study that gives preliminary empirical evidence of the value of the approach.

Acknowledgements

This research is part of the National Finnish Ontology Project (FinnONTO) 2003-2007[1], funded mainly by The National Technology Agency (Tekes) and a consortium of 37 companies and public organisations.

References

1. Harith Alani and Christopher Brewster. Ontologies and knowledge bases: Ontology ranking based on the analysis of concept structures. In *Proceedings of the 3rd international conference on Knowledge capture K-CAP 2005*, 2005.
2. N. Athanasis, V. Christophides, and D. Kotzinos. Generating on the ?y queries for the semantic web: The ics-forth graphical rql interface. In *Proceedings of the Third International Semantic Web Conference*, pages 486–501, 2004.
3. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, ACM Press, New York, 1999.
4. Tim Berners-Lee, Jim Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[1] http://www.seco.tkk.fi/projects/finnonto/

5. Michael Berry. *Survey of Text Mining Clustering, Classification, and Retrieval*. Springer-Verlag, 2004. ISBN: 978-0-387-95563-6.
6. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation 10 February 2004. Recommendation, World Wide Web Consortium, February 2004.
7. Robin Burke. Knowledge-based recommender systems. In *Burke, R.: Knowledge-based Recommender Systems. In A. Kent (ed.), Encyclopedia of Library and Information Systems. Vol. 69, Supplement 32. Marcel Dekker*, 2000.
8. Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659, 2004.
9. Klein H. K. Hirschheim, R. and K. Lyytinen. *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge University Press, Cambridge., 1995.
10. Markus Holi and Eero Hyvönen. Fuzzy view-based semantic search. In *Proceedings of the 1st Asian Semantic Web Conference (ASWC2006), Beijing, China*. Springer-Verlag, September 3-7 2006.
11. Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström, Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo, Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen. Culturesampo–finnish culture on the semantic web: The vision and first results. In *Developments in Artificial Intelligence and the Semantic Web - Proceedings of the 12th Finnish AI Conference STeP 2006*, October 26-27 2006.
12. Eero Hyvönen, Arttu Valo, Ville Komulainen, Katri Seppälä, Tomi Kauppinen, Tuukka Ruotsalo, Mirva Salminen, and Anu Ylisalmi. Finnish national ontologies for the Semantic Web - towards a content and service infrastructure. In *submitted for evaluation*, Espoo, Finland, May 2005. Helsinki University of Technology and University of Helsinki.
13. Glen Jeh and Jennifer Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543, 2002.
14. A. Maedche, S. Staab, N. Stojanovic, R. Struder, , and Y. Sure. *Semantic portal — the SEAL approach*. MIT press, Cambridge, MA, 2003.
15. David McSherry. A generalized approach to similarity-based retrieval in recommender systems. *Artificial Intelligence Review*, 18:309–341, 2002.
16. Eetu Mäkelä, Eero Hyvönen, and Samppa Saarela. Ontogator — a semantic view-based search engine service for web applications. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Nov 2006.
17. A. Rodriquez and M. Egenhofer. An asymmetric and context-dependent similarity measure. *International Journal of Geographical Information Science*, 18(3):229–256, 2004.
18. G. Rugg and P. McGeorge. The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 14(2):80–93, 1997.
19. Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report tr87-881, Cornell University Ithaca, NY, USA, 1987.
20. Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, and Dong Zhuang. Simfusion: measuring similarity using unified relationship matrix. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137, 2005.