

The Quest for Information Retrieval on the Semantic Web

David Vallet, Miriam Fernández, and Pablo Castells

Universidad Autónoma de Madrid, Escuela Politécnica Superior

Ciudad Universitaria de Cantoblanco, c/ Tomás y Valiente 11, 28049 Madrid

{david.vallet,miriam.fernandez,pablo.castells}@uam.es

Abstract. Semantic search has been one of the motivations of the Semantic Web since it was envisioned. We propose a model for the exploitation of ontology-based KBs to improve search over large document repositories. The retrieval model is based on an adaptation of the classic vector-space model, including an annotation weighting algorithm, and a ranking algorithm. Semantic search is combined with keyword-based search to achieve tolerance to KB incompleteness. Our proposal has been tested on corpora of significant size, showing promising results with respect to keyword-based search, and providing ground for further analysis and research.

Keywords: information retrieval, ontologies, semantic web, semantic search, semantic annotation

1 Introduction

Semantic search has been one of the major envisioned benefits of the Semantic Web since its emergence in the late 90's. One way to view a semantic search engine is as a tool that gets formal ontology-based queries (e.g. in RDQL, RQL, SPARQL, etc.) from a client, executes them against a knowledge base, and returns tuples of ontology values that satisfy the query [2,3,4,10].

These techniques typically use boolean search models, based on an ideal view of the information space as consisting of non-ambiguous, non-redundant, formal pieces of ontological knowledge. A knowledge item is either a correct or an incorrect answer to a given information request, thus search results are assumed to be always 100% precise, and there is no notion of approximate answer to an information need. While this conception of semantic search brings key advantages already, our work aims at taking a step beyond. In our view of Information Retrieval in the Semantic Web, a search engine returns documents, rather than (or in addition to) exact values, in response to user queries. Furthermore, as a fundamental requirement for scaling up to massive information sources, the engine should rank the documents, according to concept-based relevance criteria.

A purely boolean ontology-based retrieval model makes sense when the whole information corpus can be fully represented as an ontology-driven knowledge base. But there are well-known limits to the extent to which knowledge can be formalized this way. First, because of the huge amount of information currently available to information systems worldwide in the form of unstructured text and media documents, converting this volume of information into formal ontological knowledge at an affordable cost is currently an unsolved problem in general. Second, documents hold a value of their own, and are not equivalent to the sum of their pieces, no matter how well formalized and interlinked. Although it is useful to break documents down into smaller information units that can be reused and reassembled to serve different purposes, it is often appropriate to keep the original documents in the system. Third, wherever ontology values carry free text, boolean semantic search systems do a full-text search within the string values. If the values hold long pieces of text, a form of keyword-based search is taking place in practice beneath the ontology-based query model, whereby the “perfect match” assumption starts to become

arguable. If no clear ranking criteria is supplied, the search system may become useless if the search space is too big.

In this paper we propose an ontology-based information retrieval model meant for the exploitation of full-fledged domain ontologies and knowledge bases, to support semantic search in document repositories [16]. In contrast to boolean semantic search systems, in our perspective full documents, rather than specific ontology values from a KB, are returned in response to user information needs. To cope with large-scale information sources, we propose an adaptation of the classic vector-space model [14], suitable for an ontology-based representation, upon which a ranking algorithm is defined.

The performance of our proposed model is in direct relation with the amount and quality of information within the KB it runs upon. The latest advances in automating ontology population and semi-automatic text annotation are promising [5,9,12]. While, if ever, ontologies and meta-data (and the Semantic Web itself) become a worldwide commodity, the lack or incompleteness of available ontologies and KBs is a limitation we shall likely have to live with in the mid term. In consequence, tolerance to incomplete KBs has been set as an important requirement in our proposal.

2 State of the Art

Our view of the semantic retrieval problem is very close to the proposals in KIM [9,12]. While KIM focuses on automatic population and annotation of documents, our work focuses on the ranking algorithms for semantic search. Along with TAP [8], KIM is one of the most complete proposals reported to date, to our knowledge, for building high-quality KBs, and automatically annotating document collections at a large scale. Our work complements KIM and TAP with a

ranking algorithm specifically designed for an ontology-based retrieval model, using a semantic indexing scheme based on annotation weighting techniques.

Semantic Portals [2,3,4,10] typically provide simple search functionalities that may be better characterized as semantic data retrieval, rather than semantic information retrieval. Searches return ontology instances rather than documents, and no ranking method is provided. In some systems, links to documents that reference the instances are added in the user interface, next to each returned instance in the query answer [4], but neither the instances, nor the documents, are ranked.

The ranking problem has been taken up again in [15], and more recently [13]. Whereas both of these works are concerned with ranking query answers (i.e. ontology instances), we are concerned with ranking the documents annotated with these answers. Since our respective techniques are applied in consecutive phases of the retrieval process, it would be interesting to experiment the integration of the query result relevance function proposed by Stojanovic et al into our document relevance measures.

Finally, we share with Mayfield and Finin [11] the idea that semantic search should be a complement of keyword-based search as long as not enough ontologies and metadata are available. Also, we believe that inferencing is a useful tool to fill knowledge gaps and missing information (e.g. transitivity of the *locatedIn* relationship over geographical locations).

3 Knowledge Base and Document Base

In our view of semantic information retrieval, we assume a knowledge base has been built and associated to the information sources (the document base), by using one or several domain ontologies that describe concepts appearing in the document text. Our system can work with any

arbitrary domain ontology with essentially no restrictions, except for some minimal requirements, which basically consist of conforming to a set of root ontology classes. These are shown in Fig. 1.

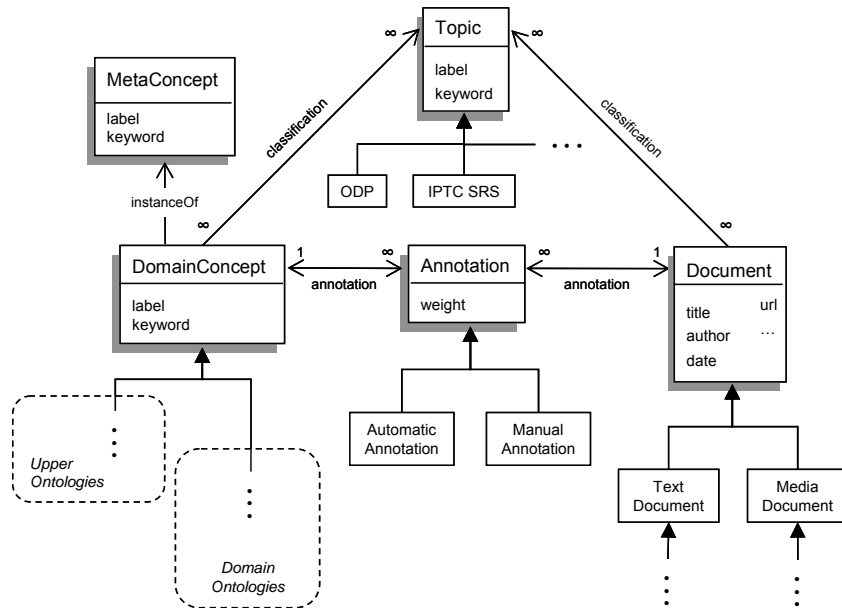


Fig. 1. Root ontology classes.

The concepts and instances in the KB are linked to the documents by means of explicit, non-embedded annotations to the documents. While we do not address here the problem of knowledge extraction from text [4,5,9,12], we provide a vocabulary and some simple mechanisms to aid in the semi-automatic annotation of documents. The automatic annotation procedure is based on a mapping of all domain concepts and instances in the KB to string keywords, similar to the ones used in other systems like KIM [9] and TAP [8]. The mapping is used by our automatic annotator to find occurrences of concepts and instances in text documents, in which case an annotation (a bi-directional link between the concept and the document) is created. Of course, further techniques are used to deal with the complexities of automatic annotation [16].

The annotations are used by the retrieval and ranking module, as will be explained in the next Section. The ranking algorithm is based on an adaptation of the classic vector-space model [14]. In the classic vector-space model, keywords appearing in a document are assigned weights reflecting that some words are better at discriminating between documents than others. Similarly, in our system, annotations are assigned a weight that reflects how important the instance is considered to be for the document meaning. Weights are computed automatically by an adaptation of the TF-IDF algorithm [14], based on the frequency of occurrence of the instances in each document. More specifically, the weight d_x of an instance x for a document d is computed as:

$$d_x = \frac{freq_{x,d}}{\max_y freq_{y,d}} \cdot \log \frac{|\mathcal{D}|}{n_x}$$

Where $freq_{x,d}$ is the number of occurrences of x in d , $\max_y freq_{y,d}$ is the frequency of the most repeated instance in d , n_x is the number of documents annotated with x , and \mathcal{D} is the set of all documents in the search space. The number of occurrences of an instance in a document is determined by using the aforementioned concept-keyword mapping. The reader is referred to [16] for further details on this.

4 Query Processing and Result Ranking

Our approach to ontology-based information retrieval can be seen as an evolution of classic keyword-based retrieval techniques, where the keyword-based index is replaced by a semantic knowledge base. The overall retrieval process is illustrated in Fig. 2. Our system takes as input a formal RDQL query. Whether this query is generated from a keyword-based query [8,13], a natural language query [4], a form-based interface [10], or more sophisticated UI techniques [5,7], is out of the focus of this paper. The RDQL query is executed against the KB, which re-

turns a list of instance tuples that satisfy the query. Finally, the documents that are annotated with these instances are retrieved, ranked, and presented to the user.

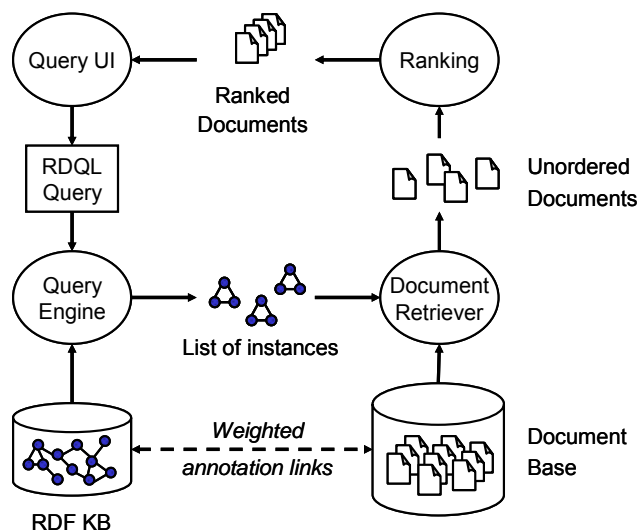


Fig. 2. Our view of ontology-based information retrieval.

The RDQL query can express conditions involving domain ontology instances and document properties (such as *author*, *date*, *publisher*, etc.). The query execution returns a set of tuples that satisfy the query. It is the document retriever's task to obtain all the documents that correspond to the instance tuples. If the tuples are only made up of instances of domain concepts, the retriever follows all outgoing annotation links from the instances, and collects all the documents in the repository that are annotated with the instances. If the tuples contain instances of document classes (because the query included direct conditions on the documents), the same procedure is followed, but restricted to the documents in the result set, instead of the whole repository. Our system uses inferencing mechanisms for implicit query expansion based on class hierarchies (e.g. organic pigments can satisfy a query for colorants), and rules such as one by which the

winner of a sports match might be inferred from the scoring. In fact, in our current implementation, it is the KB which is expanded by adding inferred statements beforehand.

Once the list of documents is formed, the search engine computes a semantic similarity value between the query and each document, as follows. Let \mathcal{O} be the set of all classes and instances in the ontology, and \mathcal{D} be the set of all documents in the search space. Let q be an RDQL query, let V_q be the set of variables in the SELECT clause of q . Let $T_q \subset \mathcal{O}^{|V_q|}$ be the list of tuples in the query result set, where for each tuple $t \in T_q$ and each $v \in V_q$, $t_v \in \mathcal{O}$.

We represent each document in the search space as a *document vector* $d \in \mathcal{D}$, where d_x is the weight of the annotation of the document with concept x for each $x \in \mathcal{O}$, if such annotation exists, and zero otherwise. We define the *extended query vector* q as given by $q_x = \left| \left\{ v \in V_q \mid \exists t \in T_q, t_v = x \right\} \right|$, i.e. the query vector coordinate corresponding to x is the number of variables in the RDQL query for which there is a tuple t where the variable is instantiated by x . If x does not appear in any tuple, we assign $q_x = 0$. Now, the similarity measure between a document d and the query q is computed as:¹

$$\text{sim}(d, q) = \frac{d \cdot q}{|d| \cdot |q|}$$

Where the knowledge in the KB is incomplete, the semantic ranking algorithm performs very poorly: RDQL queries will return less results than expected, and the relevant documents will not be retrieved, or will get a much lower similarity value than they should. As limited as

might be, keyword-based search may perform better in these cases. To cope with this, our ranking model combines the semantic similarity measure with the similarity measure of a keyword-based algorithm. The final value for ranking is computed as $s \cdot \text{sim}(d,q) + (1 - s) \text{ksim}(d,q)$, where ksim is computed by a keyword-based algorithm. We have taken $s = 0.5$, which seems to perform well in our experiments.

5 Experimental Testing

We have tested our system on a corpus of 145,316 documents from the CNN web site.² We have used the KIM domain ontology and KB [9], publicly available as part of the KIM Platform, developed by Ontotext Lab,³ with minor extensions and adjustments to conform to our top-level ontology meta-model. We have also manually added classes and instances in areas where the KIM KB fell short (such as the Sports domain), in order to support a larger test bed for experimentation. Our current implementation is compatible with both RDF and OWL. The complete KB includes 281 classes, 138 properties, 35,689 instances, and 465,848 sentences, stored on a MySQL back-end using Jena 2.2. Based on the concept-keyword mapping available in the KIM KB, over $3 \cdot 10^6$ annotations are automatically generated by the procedure mentioned in Section 3.

We have tested the retrieval algorithm on a set of examples, and compared it to a keyword-only search, using the Jakarta Lucene library.⁴ Fig. 3 shows an average comparison of the per-

¹ For the sake of conciseness, we are omitting here certain minor details, such as normalization factors, correction functions, etc., for the optimization of the algorithm.

² http://dmoz.org/News/Online_Archives/CNN.com

³ <http://www.ontotext.com/kim>

⁴ <http://lucene.apache.org>

formance of our system over a set of twenty queries, such as “banks that trade on NASDAQ, with fiscal net income greater than two billion dollars,” and similar ones.

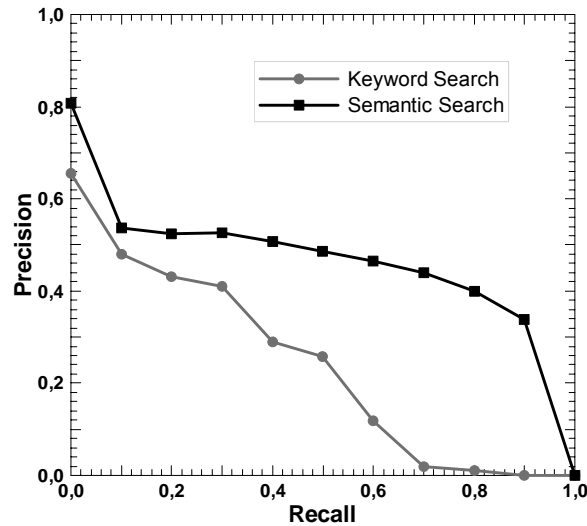


Fig. 3. Average precision vs. recall curve for a test bed of twenty queries.

6 Discussion

The added value of semantic information retrieval with respect to traditional keyword-based retrieval, as envisioned in our approach, relies on the additional explicit information – type, structure, hierarchy, relations, and rules, on the concepts referenced in the documents, represented in an ontology-based KB, as opposed to classic flat keyword-based indices. Semantic search introduces an additional step with respect to classic information retrieval models: instead of a simple keyword index lookup, the semantic search system processes a semantic query against the KB, which returns a set of instances. This can be seen as a form of query expansion, where the set of instances represent a new set of query terms, leading to higher recall values. Further implicit query expansion is achieved by inference rules, and exploiting class hierarchies. The rich con-

cept descriptions in the KB provide useful information for disambiguating the meaning of documents.

In summary, our proposal achieves the following improvements with respect to keyword-based search:

- Better recall when querying for class instances. For example, querying for “British companies quoted on NYSE” would return documents that mention e.g. *Barclays PLC*, *Vodafone* and other such companies, even if the words “British” and “NYSE” are not present in the documents.
- Better precision by using structured semantic queries. Structured queries allow expressing more precise information needs, leading to more accurate answers. For instance, in a keyword-based system, it is not possible to distinguish a query for USA players in European basket teams vs. European players in USA teams, which is possible with a semantic query.
- Better precision by using query weights. Variables with low weights are only used to impose conditions on the variables which really matter. For example, the user can search for news about USA players in European teams, regardless of whether the news mention the team at all, or the nationality of the player.
- Better recall by using class hierarchies and rules. For example, a query for *WaterSports* in *Spain* would return results in *ScubaDiving*, *Windsurf*, and other subclasses, in *Cádiz*, *Málaga*, *Almería*, and other Spanish locations (by transitivity of *locatedIn*).
- Despite the separation of the content space (documents) and the concept space, it is possible to combine conditions on concepts and conditions on contents. For example, in the query “film reviews published within the current year about Japanese sci-fi movies”, the type

(film review) and date (current year) refer to the document, whereas the rest of the query defines conditions on some concept (a movie), not in the document space, that annotates the document.

- The improvements of our method with respect to keyword-based search increase with the number of clauses in (i.e. the specificity of) the formal query. This is not surprising, since the higher the complexity of the information need, the more query information is lost in a keyword-based query.
- The degree of improvement of our semantic retrieval model depends on the completeness and quality of the ontology, the KB, and the concept labels. For the sake of robustness, the system resorts to keyword-based search when the KB returns poor results.

The combination of keyword ranking and semantic ranking is tricky. We have observed that occasionally a good semantic ranking score is spoiled by a low keyword-based value. A simple solution would be to set a minimum threshold for the keyword-based score to be counted. Anyhow, these cases, albeit infrequent, suggest that more sophisticated methods than the linear combination of both values should be researched to improve our initial results.

7 Conclusion

Our approach can be seen as an evolution of the classic vector-space model, where keyword-based indices are replaced by an ontology-based KB, and a semi-automatic document annotation and weighting procedure is the equivalent of the keyword extraction and indexing process. We show that it is possible to develop a consistent ranking algorithm on this basis, yielding measurable improvements with respect to keyword-based search, subject to the quality and critical mass

of metadata. Our proposal inherits all the well-known problems of building and sharing well-defined ontologies, populating knowledge bases, and mapping keywords to concepts. Recent research on these areas is yielding promising results [5,9]. It is our aim to provide a consistent model by which any advancement on these problems is played to the benefit of semantic search improvements.

There is ample room for further improvement and research beyond our current results. For instance, our annotation weighting scheme is not taking advantage yet of the different relevance of structured document fields (e.g. title is more important than body). Annotating documents with statements, besides instances, is another interesting possibility to experiment with. Also, we are currently extending our model with a profile of user interests for personalized search [1].

8 References

1. Castells, P., Fernández, M., Vallet, D., Mylonas, P., Avrithis, Y.: Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. 1st IFIP International Workshop on Web Semantics (SWWS 2005). LNCS Vol. 3532 (2005) 455-470
2. Castells, P., Foncillas, B., Lara, R., Rico, M., Alonso, J. L.: Semantic Web Technologies for Economic and Financial Information Management. 1st European Semantic Web Symposium (ESWS 2004). LNCS Vol. 3053 (2004) 473-487
3. Castells, P., Perdrix, F., Pulido, E., Rico, M., Benjamins, V. R., Contreras, J., Lorés, J.: Neptuno: Semantic Web Technologies for a Digital Newspaper Archive. 1st European Semantic Web Symposium (ESWS 2004). LNCS Vol. 3053 (2004) 445-458

4. Contreras, J., Benjamins, V. R., et al: A Semantic Portal for the International Affairs Sector. 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004). LNCS Vol. 3257 (2004) 203-215
5. Dill, S., et al: A Case for Automated Large Scale Semantic Annotation. Journal of Web Semantics 1:1 (2003) 115-132
6. García-Barriocanal, E., Sicilia, M.A.: User Interface Tactics in Ontology-Based Information Seeking. Psychology e-journal 1:3 (2003) 243-256
7. Guarino, N., Masolo, C., and Vetere, G.: OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems 14:3 (1999) 70-80
8. Guha, R. V., McCool, R., and Miller, E.: Semantic search. 12th International World Wide Web Conference (WWW 2003). Budapest, Hungary (2003) 700-709
9. Kiryakov, A., Popov, B., Terziev, I., Manov, Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Semantics 2:1 (2004) 49-79
10. Maedche, A., Staab, S., Stojanovic, N., Studer, R., Sure, Y.: SEMantic portAL: The SEAL Approach. In: Fensel, D., Hendler, J. A., Lieberman, H., Wahlster, W. (eds.): Spinning the Semantic Web. MIT Press, Cambridge London (2003) 317-359
11. Mayfield, J., Finin, T.: Information retrieval on the Semantic Web: Integrating inference and retrieval. Workshop on the Semantic Web at the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003). Toronto, Canada (2003)
12. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: KIM – A Semantic Platform for Information Extaction and Retrieval. Journal of Natural Language Engineering 10:3-4 (2004) 375-392

13. Rocha, C., Schwabe, D., de Aragão, M. P.: A Hybrid Approach for Searching in the Semantic Web. International World Wide Web Conference (WWW 2004), New York (2004) 374-383
14. Salton, G., McGill, M. Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
15. Stojanovic, N., Studer, R., Stojanovic, L.: An Approach for the Ranking of Query Results in the Semantic Web. 2nd International Semantic Web Conference (ISWC 2003). LNCS Vol. 2870 (2003) 500-516
16. Vallet, D., Fernández, M., Castells, P.: An Ontology-Based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC 2005). LNCS Vol. 3532 (2005) 455-470

David Vallet is Research Assistant at the Universidad Autónoma de Madrid (UAM), where he earned a MS degree in Computer Science. His research interests are focused on the confluence of Information Retrieval, User Modeling, and Context Modeling.

Miriam Fernández holds a MS degree in Computer Science from UAM, where she is a Research Assistant. Her research interests include Ontology Engineering, Semantic Search, and Semantic Annotation.

Pablo Castells is Associate Professor at the UAM since 1999. He earned a PhD in Computer Science in 1994 at UAM, with a thesis on Automated Theorem Proving. In 1994/95 he was a postdoctoral research fellow at the University of Southern California. More recently, he has led or participated in several national and international projects in the areas of the Semantic Web and Knowledge-Based Systems, in application domains such as the News, Finance, and Healthcare. His current research focuses on Information Retrieval, Personalization technologies, and Semantic Web Services.