Semantically enhanced Information Retrieval: an ontology-based approach

Miriam Fernández[1], Iván Cantador[2], Vanesa López[1], David Vallet[2], Pablo Castells[2], Enrico Motta[1]

[1] *Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom*

[2] *Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Madrid, Spain*

Summary

Currently, techniques for content description and query processing in Information Retrieval (IR) are based on keywords, and therefore provide limited capabilities to capture the conceptualizations associated with user needs and contents. Aiming to solve the limitations of keyword-based models, the idea of conceptual search, understood as searching by meanings rather than literal strings, has been the focus of a wide body of research in the IR field. More recently, it has been used as a prototypical scenario (or even envisioned as a potential "killer app") in the Semantic Web (SW) vision, since its emergence in the late nineties. However, current approaches to semantic search developed in the SW area have not yet taken full advantage of the acquired knowledge, accumulated experience, and technological sophistication achieved through several decades of work in the IR field. Starting from this position, this work investigates the definition of an ontology-based IR model, oriented to the exploitation of domain Knowledge Bases to support semantic search capabilities in large document repositories, stressing on the one hand the use of fully-fledged ontologies in the semantic-based perspective, and on the other hand the consideration of unstructured content as the target search space. The major contribution of this work is an innovative, comprehensive semantic search model, which extends the classic IR model, addresses the challenges of the massive and heterogeneous Web environment, and integrates the benefits of both keyword and semantic-based search. Additional contributions include: an innovative rank fusion technique that minimizes the undesired effects of knowledge sparseness on the yet juvenile SW, and the creation of a large-scale evaluation benchmark, based on TREC IR evaluation standards, which allows a rigorous comparison between IR and SW approaches. Conducted experiments show that our semantic search model obtain comparable and better performance results (in terms of MAP and P@10 values) than the best TREC automatic system.

Keywords: Semantic Web, Information Retrieval, semantic search

1 INTRODUCTION

1.1 Motivation

With the continued growth of online information, the processes of searching and managing massive scale content have become increasingly challenging, bringing along the upsurge of huge new markets. Major search engines, like Google[1], Yahoo![2] or Bing[3], are constantly introducing new features to improve users' search experience, including the introduction of novel mechanisms to handle multimedia content[4]; the categorization of information sources such as news, blogs, forums or books[5]; the introduction of metadata by publishers to enhance the visualization of results[6]; or the use of personal and contextual information, such as social networks, location, etc., to particularize results according to users' tastes, interests and situations[7].

Even though search engine technology has experienced impressive enhancements in the last decade, the content description and query processing techniques Information Retrieval (IR) technology currently builds upon are still mostly based on keywords, and therefore provide limited capabilities to capture and exploit the conceptualizations involved in user needs and content meanings. For instance, limitations include the inability to account for relations between search terms (e.g., "hurricanes originated in Mexico" vs. "hurricanes that have affected Mexico", "books about recommender systems" vs. "systems that recommend books"), to handle searches that involve a secondary sense of a term (e.g. "Victor Valdés", the goal keeper vs. "Victor Valdés", the video processing researcher) or to integrate information distributed over several Web resources, (e.g. searches regarding products or services).

Aiming to solve the limitations of keyword-based models, the idea of semantic search, understood as searching by meanings rather than literal strings, has been the focus of a wide body of research in the Information Retrieval (IR) and the Semantic Web (SW) communities. However, these two fields have had a different understanding of the problem.

[1] Google search engine, http://www.google.com/
[2] Yahoo! search engine, http://www.yahoo.com/
[3] Microsoft Bing search engine, http://www.bing.com/
[4] Google images, http://images.google.com/,
Yahoo! images, http://images.search.yahoo.com/,
YouTube, http://www.youtube.com/,
Yahoo! videos, http://video.yahoo.com/
[5] http://googleblog.blogspot.com/2009/10/refine-your-search-results-with-new.html
[6] http://developer.yahoo.com/searchmonkey/
[7] http://www.google.com/ig/

Semantic search has been present in the IR field since the early eighties (Croft, 1986), if not earlier (Van Rijsbergen, 1979). Some of these approaches are based on statistical methods that study the co-occurrence of terms (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990; Dumais, 1990), and therefore they capture and exploit rough and fuzzy conceptualizations. Other IR approaches apply linguistic algorithms (Gonzalo, Verdejo, Chugur & Cigarrán, 1998), modelled on human language processing structures and mechanisms, but rely on thesauri and taxonomies, where the level of conceptualization is often shallow and sparse, especially at the level of relations, which are commonly at the core of expressing user needs and finding the answers.

On the other hand, semantic search can be said to have become one of the "philosopher's stones" in the SW community since its emergence in the late nineties. The SW vision was brought about with the aim of helping automate tasks that require a certain level of conceptual understanding of the objects involved (e.g., information objects) or the task itself, and enabling software programs to automatically find and combine information and resources in consistent ways. At the core of these new technologies, ontologies (Gruber, 1993) were envisioned as key elements to represent knowledge that could be understood, used and shared among distributed applications and agents. Their potential to overcome the limitations of keyword-based search in the IR context was soon envisaged, and was explored by several researchers in the SW area (Maedche, Staab, Stojanovic, Studer & Sure, 2003; Guha, McCool & Miller, 2003; Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004). However, these approaches exhibit certain limitations like: a) the still sparseness of the available SW content (Sabou, Gracia, Angeletou, d'Aquin & Motta, 2007), leading to knowledge incompleteness when applying search to heterogeneous sources of information, b) the poor usability of the systems, specially at the level of query, requiring users to manage complex languages or interfaces to express their information needs, c) the lack of ranking algorithms to cope with large-scale information sources, etc (see Section 2.3). One may say that the undertakings in information search and retrieval from the SW community have not yet taken full advantage of the acquired knowledge, accumulated experience, and theoretical and technical achievements developed through several decades of work in the IR field tradition.

Starting from this position, and aiming to bridge the gap between these two communities, this work investigates the definition of an ontology-based IR model, oriented to the exploitation of domain Knowledge Bases (KBs) to support semantic search capabilities in large document repositories, stressing on the one hand the use of fully-fledged ontologies in the semantic-based perspective, and on the other the consideration of unstructured content as the target search space. In other words, this work explores the use of semantic information to support more expressive queries and more accurate results, while the retrieval problem is formulated in a way that is consistent with the IR field, thus drawing benefit from the state of the art in this area, and enabling more realistic and applicable approaches.

1.2 Contributions

Our contributions fall into four major categories:

- **Better understanding of the semantic search problem, the potential of semantic enhancements in IR technology, the current achievements from the IR and SW fields, and the fundamental differences between both perspectives**. Despite the large amount of work on conceptual search in the IR field, semantic search has been addressed as a refinement or smooth extension of traditional IR techniques rather than as a radical new paradigm, until the emergence of the SW. We study the strengths and weaknesses of the proposals towards the semantic search paradigm from both fields.
- **Definition and realization of a novel semantic retrieval model**. In order to address the shortcomings in prior semantic search approaches, this work proposes the exploitation of fine-grained domain ontologies and KBs to improve semantic retrieval in large repositories of unstructured information, extending the general ontology-based search capabilities towards more widely applicable IR-oriented search capabilities.
- **Investigate the feasibility of semantic retrieval in the Web environment**. As a step towards a proof of concept of the feasibility of semantic retrieval within large-scale and heterogeneous environments, the proposed model is modified to address scalability, heterogeneity and usability challenges.
- **Creation of semantic retrieval evaluation benchmarks**. The standardization of experimental practice in keyword-based IR has come a long way. In contrast, there is not an equivalent body of methodologies and datasets for the evaluation of semantic retrieval models. This work aims to take a step forward, starting from traditional IR evaluation measures and datasets to provide evaluation benchmarks for ontology-based retrieval technologies.

1.3 Structure of the paper

The rest of the paper is organized as follows. Section 2 describes related work in both the IR and SW areas, and addresses a common understanding of what semantic search is, and where we are standing in the progress towards semantic information retrieval. Section 3 presents a semantic search approach that combines, under a common model, the main achievements in semantic search from the IR and SW perspectives. Section 4 presents the research done to scale the above model to an open, massive and heterogeneous environment such as the Web. The evaluation of the Web extended model is reported in section 5. Conclusions and future work are presented in section 6.

2 RELATED WORK

Any IR system is based on a logic representation of user information needs, and the information supplied by the information objects in the search space, in such a way that the comparison between queries and potential answers takes place in the ideal model. The various logic representations proposed in the area (Lewis & Gale, 1994) respond, on the one hand, to the requirement of being efficiently processable by an IR system, and necessarily entail some information loss. This is clear, for instance, in the representation of

information needs by a simple list of keywords, as is the case in currently dominant paradigms in both research and industry.

An important aspect of semantic search approaches is that practically all of them use conceptual representations of content beyond plain keywords, and many of them also attempt to provide conceptual representations of user needs, as a way to enhance mainstream IR technologies.

2.1 Semantic Search: an IR perspective

The elaboration of conceptual frameworks and their introduction in IR models have wide precedents. For instance, (Croft, 1986) proposed a representation where domain knowledge is modelled by a **thesaurus** of concepts, each one having a name, some relations to other concepts, and a list of more or less ad-hoc rules (defined on a per-case basis) to recognize the concepts in a textual passage. The considered relations between concepts included synonymy, hyponymy and instantiation, meronymy and similarity. These concepts and relations are used to expand both queries and document indexing entries. Aware of the cost of producing domain knowledge, Croft suggested using such knowledge as an enabler of incremental improvement over purely statistical methods, in such a way that the performance of the latter is retained in the absence or incompleteness of the former.

Croft's work is representative of a trend which, during the same period, attempted to enhance the performance of IR systems by strengthening content representation through the use of conceptual abstractions. In this line, and possibly under the influence of knowledge based systems in the Artificial Intelligence area, several approaches in the eighties investigated the use of **semantic networks** to enrich the representation of the indexing terms (Cohen & Kjeldsen, 1987; Shoval, 1981).

The idea of augmenting the semantic representation of a document beyond a set of plain words is in fact present in earlier works to those decades, such as Karen Spärck Jones' PhD thesis (Spärck Jones, 1964). In it, the author reflects on the flexible, non univocal correspondence between words and meanings, and the role of relations between words (synonymy, antonymy, hyponymy, entailment, and others) in the description of meanings. Her work considers the notion of predefined semantic primitives, consisting in essence of (domain-specific or generic) concepts taken from a thesaurus (the Roget's (Lloyd & Roget, 1982)), which are automatically extended with emergent semantic entities, observable in the analysis of a text corpus.

Considerable research followed in which several authors have kept progressing on conceptual approaches to IR based on domain knowledge. One of the pursued lines in this direction is the one based on **linguistic approaches**, among which the use of resources like WordNet[8] is particularly representative of the use of explicit conceptual descriptions (Vorhees, 1994; Madala, Takenobu & Hozumi, 1998).

Beyond WordNet, or complementarily to its use, many works have explored the use of thesauri with a lower or higher specialization level to introduce enhancements in search effectiveness (Harbourt, Syed, Hole & Kingsland, 1993; Hersh & Greenes, 1990). One of the most common uses of thesauri in this context is the expansion of query terms, based on the mapping of query words to thesauri elements, and the extension of the latter through their relations to other terms in the thesauri.

From a very different starting point, the idea of raising IR techniques to a higher conceptual level is also present in **Latent Semantic Analysis** (LSA) techniques, widely studied and applied in diverse domains (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990). As distinct from thesauri-oriented techniques, concepts emerge in LSA by means of algebraic methods, based on the frequency of words in the documents of a corpus.

2.2 Semantic search: a SW perspective

The introduction of ontologies to move beyond the capabilities of current search technologies has been an often portrayed scenario in the area of semantic-based technologies since the late nineties (Luke, Spector & Rager, 1996). Compared to what is usual in thesauri, the emphasis on formalization is much higher in ontologies, which seek to describe the world (or at least a domain) on the basis of a descriptive logic that axiomatizes the ontology classes, their relations, and the properties of both (symmetry, transitivity, equivalences, etc.) in suitable terms to be formally reasoned upon.

In contrast with the standard IR model, a number of systems referred to as "semantic search systems" in the SW area, provide search mechanisms over a single KB rather than documents. Hence here the emphasis is on developing mechanisms that are able to capture user queries and convert them to a formal query representation, e.g. SPARQL. In general, this vision makes sense when the whole information corpus can be fully represented as a formal KB. But there are limits to the extent to which knowledge can be formalized in this way. The so-called semantic portals (Contreras, et al., 2004; Maedche, Staab, Stojanovic, Studer & Sure, 2003) and ontology-based Question Answering systems (Bernstein & Kaufmann, 2006; Cimiano, Haase & Heizmann, 2007) are examples of this approach.

There are nonetheless approaches in this context that explicitly consider keeping, along with the domain ontologies and KBs, the original documents in the retrieval model, where the relations between ontologies and documents are established by annotation relations. In this line, KIM (Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004; Popov, Kiryakov, Ognyanoff, Manov & Kirilov, 2004), TAP (Guha, McCool & Miller, 2003) and more recently, Hakia[9], are examples of wide-ranging achievements on the construction of high-quality KBs, and the automatic annotation of documents on a large scale.

While these approaches are focused on knowledge extraction from text, recent solutions like Powerset[10] aim to exploit existing and publicly available metadata, like the one generated as part of the Linked Open Data (LOD) initiative[11], and provide this knowledge in combination with textual documents. In the particular case of Powerset, FreeBase[12] and Wikipedia[13] are integrated as the main metadata and textual information sources respectively.

[8] http://wordnet.princeton.edu/

[9] http://www.hakia.com/

[10] http://www.powerset.com/

[11] http://linkeddata.org/

[12] http://www.freebase.com/

Finally, an interesting trend of semantic search in recent years is the use of explicit metadata provided by publishers. These metadata are embedded in Web pages using RDFa[14], or Microformats[15] and exploited by commercial search engines, like Yahoo! SearchMonkey[16], or Google Rich Snippets[17], to enhance the visualization of results.

2.3 Classification and limitations of semantic search approaches

The classification of semantic search approaches is complex, not just because of their diversity, in the sense of how differently this problem has been approached in the literature, but also because of the large number of dimensions involved in the information search task. This section proposes a set of general criteria under which SW and IR approaches can be classified and compared, identifying their key advantages and limitations.

[Table 1 about here]

The classification criteria are summarized in Table 1 and comprise:

Semantic knowledge representation: three main trends can be distinguished in the literature based on the type and use of semantic knowledge representation: a) *statistical approaches*, like LSA (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990), use statistical models to identify groups of words that commonly appear together, and therefore may jointly describe a particular reality; b) *linguistic conceptualization approaches* (Gonzalo, Verdejo, Chugur & Cigarrán, 1998; Madala, Takenobu & Hozumi, 1998; Giunchiglia, Kharkevich & Zaihrayeu, 2009) are based on light conceptualizations, usually considering few types of relations between concepts, and low information specificity levels, and; c) *ontology-based proposals* (Popov, Kiryakov, Ognyanoff, Manov & Kirilov, 2004; Guha, McCool & Miller, 2003) consider a much more detailed and densely populated conceptual space in the form of ontology-based KBs.

Scope: semantic search has been applied in different environments such as the *Web* (Finin, Mayfield, Fink, Joshi & Cost, 2005; Fernández et al., 2008), *controlled repositories* (Popov, Kiryakov, Ognyanoff, Manov & Kirilov, 2004), or even the *desktop* (Chirita, Gavriloaie, Ghita, Nejdl & Paiu, 2005). Obtaining conceptualizations to cover the meanings involved in all Web content as well as the automatic annotation of these conceptualizations with some degree of completeness is still an open challenge. Restricting themselves to more reduced environments, many systems have been developed and tested over controlled repositories, where the available information is enclosed in one or few domains of knowledge. In a third degree of complexity, the desktop environment provides easier ways to extract the semantic information from semi-structured contents such as e-mails, folders, etc. Some works do not explicitly state their potential or limitations in scope and scale, but the considerable computational complexity involved in their methods (see e.g. Giunchiglia, Kharkevich & Zaihrayeu, 2009) leaves scalability (in particular, Web scalability) as a non addressed issue.

Query: another relevant aspect that characterizes semantic search models is the way the user expresses his information needs. Four different approaches can be identified in the state of the art, according to a gradual increase of their level of formality and usage complexity. At a first level, queries are expressed by means of *keywords* (Guha, McCool & Miller, 2003). This is the most traditional way of consultation, but also the least expressive one, since the information need is represented as a set of terms without any explicit relation between them. A second level involves a *natural language* representation of the information need (López, Sabou, Uren & Motta, 2009). This kind of query provides more information than the keyword-based approach since a linguistic analysis can be performed to extract syntactic information, such as subject, predicate, object and other details of the sentence. A third level in formality is portrayed by *controlled natural language* systems (Bernstein & Kaufmann, 2006; Cohen, Mamou, Kanza & Sagiv, 2003; Giunchiglia, Kharkevich & Zaihrayeu, 2009), where the query may be expressed by adding tags that represent properties, values or objects within the consultation or by more sophisticated methods like (Giunchiglia, Kharkevich & Zaihrayeu, 2009), which builds a logic-based approach on top of WordNet synsets. These types of queries can be more easily processed and mapped to the corresponding classes, properties and values of a schema or ontology describing the search space, thus facilitating the acquisition of semantically related information. Finally, the most formal ontology-based search systems *use ontology-query languages* such as RDQL (Seaborne, 2004), SPARQL (Prud'hommeaux & Seaborne, 2006), etc. The full expressive power of this kind of query allows the system to automatically retrieve in a highly precise way the information that satisfies the user's need. These systems, which demand a high formalization of queries, tend to be impractical from a usability point of view. On the other hand, it can be argued that increasing the expressivity of queries helps to improve the quality of results, since the returned results must strictly hold all the conditions of the formal query and therefore, they are assumed to be 100% precise. A trade-off between usability and query expressivity should be achieved, bringing an inherent degree of fuzziness during the data search process.

Content retrieved: semantic retrieval approaches can be characterized by whether they aim at data or information retrieval. While the majority of IR approaches return documents as response to user requests, and therefore should be classified as *information retrieval models*, a large amount of ontology-based approaches return ontology instances rather than documents, and therefore may be classified as *data retrieval models*. A data retrieval model makes sense when the whole information corpus can be fully represented as a KB. However, converting the huge amount of information available worldwide, in the form of unstructured text and media documents, into formally characterized knowledge at an affordable cost is currently an unsolved problem.

Content ranking: While IR approaches have traditionally addressed the ranking of documents, most ontology-based approaches do not consider ranking query results in general, or base their ranking functionality on traditional keyword-based approaches (Guha, McCool & Miller, 2003). A few approaches take advantage of semantic information to generate query result rankings, but generally KB instances rather than documents are ranked (Stojanovic, 2003). These methodologies are not yet adapted to large and

[13] http://www.wikipedia.org/
[14] http://http://www.w3.org/TR/xhtml-rdfa-primer/
[15] http://microformats.org/
[16] http://developer.yahoo.com/searchmonkey/
[17] http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html

heterogeneous environments (e.g., the Web) where the majority of content is still unstructured.

The set of limitations associated to each of the previously mentioned classification criteria is summarized in Table 2.

[Table 2 about here]

As shown in Table 2, out of the selected criteria, two additional major open issues in ontology-based search approaches can be pointed out.

- *The problem of knowledge incompleteness*: the difficulties and cost of building and maintaining rich semantic resources is a well-known fundamental hurdle, already identified by the earliest works in the field (Croft, 1986). A fundamental issue here is to discern what level of detail (depth) and coverage (breadth) is appropriate, and how well we may cope with the remaining incompleteness beyond that point. A potential way to satisfy the latter is by means of a graceful degradation to a classic IR system which gets by without semantics when there is insufficient domain knowledge.
- *The problem of evaluating semantic search models*: while IR systems traditionally compete against each other under formal evaluation frameworks, e.g. at the annual TREC conference[18], or using published datasets, there are no standard evaluation measures or benchmarks for ontology-based retrieval and, furthermore, there is not a well established evaluation methodology. To the best of our knowledge, none of the ontology-based retrieval approaches reported in the literature have been validated in such rigorous ways. A partial exception is recent work by (Giunchiglia, Kharkevich & Zaihrayeu, 2009), although the work is focused on Wordnet, thus leaving the integration of domain ontologies with TREC as an open problem, and does not report improvements with respect to the performance of the best TREC systems.

3 AN ONTOLOGY-BASED IR MODEL

Our approach builds upon principles from (Castells, Fernández & Vallet, 2007), where a general framework to leverage ontologies in the frame of a traditional vector space IR model is developed. In our present work, we address the further challenges involved in making the approach feasible on large and heterogeneous information repositories, as required to target practical and realistic settings such as the Web. Furthermore, we seek to devise a methodological approach supporting a formal evaluation of the ontology-based search approach in the spirit and standards of conventional IR practice.

The proposed extensions over an ontology-based IR model and the complete evaluation of the model are presented in sections 4 and 5 respectively. In this section, we provide a brief overview of the original base model, which provides the ground foundation for the research presented herein. For specific details about this model and its evaluation, see (Castells, Fernández & Vallet, 2007).

[18] Text REtrieval Conference (TREC), http://trec.nist.gov/

The core semantic search model is based on an adaptation of the classic keyword-based IR model (Baeza & Ribeiro, 1999). It spans the four main processes of an IR system: indexing, querying, searching and ranking (Figure 1). However, as opposed to traditional keyword-based IR models, in our approach, the query is expressed in terms of an ontology-based query language (SPARQL), and the external resources used for indexing and query processing consist of an ontology and its corresponding KB. The indexing process is equivalent to a semantic annotation process. Instead of creating an inverted index where the keywords are associated with the documents where they appear, in the case of our ontology-based IR model, the inverted index contains semantic entities (meanings) associate to the documents where they appear. The relation or association between a semantic entity and a document is what we call annotation.

[Figure 1 about here]

The overall retrieval process consists of the following steps:

1. The system takes as input a formal SPARQL query.
2. The SPARQL query is executed against a KB, returning a list of semantic entities that satisfy the query. This process is purely Boolean (i.e., based on an exact match), so that the returned instances must strictly hold all the conditions of the formal query.
3. The documents that are annotated (indexed) with the above instances are retrieved, ranked, and presented to the user. In contrast to the previous phase, the document retrieval phase is based on an approximate match, since the relation between a document and the concepts that annotate it has an inherent degree of fuzziness.

The steps listed above are described in more detail in the following subsections, from indexing to query processing, document retrieval and ranking.

3.1 Semantic indexing

In our view of semantic IR, it is assumed that a KB has been built and associated to the information sources (the document base), by using one or several domain ontologies that describe concepts appearing in a document text. The concepts and instances in the KB are linked to the documents by means of explicit, non-embedded annotations of the documents. Since we do not address the problem of knowledge extraction from text (Contreras, et al., 2004; Dill, et al., 2003; Handschuh, Staab & Ciravegna, 2002; Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004; Popov, Kiryakov, Ognyanoff, Manov & Kirilov, 2004), we provide a vocabulary and some simple mechanisms to aid in the semi-automatic annotation of documents, once ontology instances have been created (manually or automatically).

These annotations are later used during the retrieval and ranking processes. As we shall describe in the next subsection, the ranking algorithm is based on an adaptation of the classic IR vector space model (Salton, 1986). In this model, keywords appearing in a document are assigned weights reflecting the fact that some words are better at discriminating between documents than others. Similarly, in our system, annotations are assigned weights that reflect the discriminative power of instances with respect to the documents. Weights are computed automatically by an adaptation of the TF-IDF

algorithm (Salton, 1986), 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:

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

where $freq_{x,d}$ is the number of occurrences in d of the keywords attached to x, $max_y\ freq_{y,d}$ is the frequency of the most repeated instance in d, n_x is the number of documents annotated with x, and D is the set of all documents in the search space.

3.2 Querying, searching and ranking

The query execution returns a set of tuples that satisfy the SPARQL query. We then extract the semantic entities from those tuples and access the semantic index to collect all the documents in the repository that are annotated with these semantic entities. Once the list of documents is formed, the search engine computes a semantic similarity value between the query and each document, using an adaptation of the classic vector space IR model.

[Figure 2 about here]

As shown in Figure 2, each document in the search space is represented as a document vector where each element corresponds to a semantic entity. The value of an element is the weight of the annotation between the document and the semantic entity, if such annotation exists, and zero otherwise. The query vector is generated weighting the variables in the SELECT clause of the SPARQL query. For testing purposes, the weight of each variable of the query was set to 1, but in the original model, users are allowed to manually set this weight according to their interest. Once the vectors are constructed, the similarity measure between a document d and the query q is computed as:

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

3.3 Dealing with the problem of knowledge incompleteness: rank fusion

If the knowledge in the KB is incomplete (e.g., there are documents about travel offers in the knowledge source, but the corresponding instances are missing in the KB), the semantic ranking algorithm performs very poorly: SPARQL 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 will likely perform better in these cases. To cope with this, our ranking function combines the semantic similarity measure with the similarity measure of a keyword-based algorithm.

Combining the output of several search engines has been a widely addressed research topic in the IR field (Croft, 2000; Lee, 1997). After testing several approaches, we selected the so-called CombSUM strategy (Shaw & Fox, 1993), which has been found to be among the most simple and effective in prior works, and consists of computing the combined ranking score by a linear combination of the inputs. That is, in our case the final score is $\lambda \cdot sim(d,q) + (1 - \lambda) \cdot ksim(d, q)$, where *ksim* is computed by a keyword-based algorithm, and $\lambda \in [0,1]$. We set $\lambda = 0.5$, which seemed to perform well in our experiments. Obviously, for the combination of scores to make sense, the scores have to be first made comparable, which involves a normalization step. For this purpose, we use our own optimized normalization method (Fernández, Vallet & Castells, 2006), which not only scales the scores to the same range (the [0,1] range) as other standard approaches proposed in the literature do (Lee, 1997), but also undoes potential biases in the distribution of the scores.

4 SEMANTIC RETRIEVAL ON THE WEB

The semantic search model detailed in Section 3, as well as other semantic approaches that have proved to work well in specific domains (Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004; Chirita, Gavriloaie, Ghita, Nejdl & Paiu, 2005), still have to undertake further steps towards an effective deployment of semantic search on a decentralized, heterogeneous, dynamic and massive repository of content such as the Web. Accomplishing this objective involves tackling several problems such as:

- **Heterogeneity**. The experiments described in (Castells, Fernández & Vallet, 2007) are based on the KIM KB. This ontology provides a reasonably good coverage of knowledge areas of general importance (geographical locations, organizations, etc.). Nonetheless, the contents available on the Web range over a potentially unlimited number of domains. Therefore, substantially better means to procure proper knowledge coverage levels are required. To address this problem we propose: a) the generation of a SW gateway that provides access to large amounts of online available semantic metadata (Section 4.4) covering a significant number of domains, and, b) the adaptation of the previous model to exploit the semantic information provided by the SW gateway. This information is used at indexing time (Section 4.1) and at query time (Section 4.2), to improve the domain coverage.

- **Scalability**. Scalability issues are still a pervading open problem in ontology-based technologies. A popular example of this is Powerset[19], whose coverage is limited to Wikipedia. Scaling our model to the Web environment implies, on the one hand, to exploit all the increasing available semantic metadata in order to provide a good coverage of topics and, on the other hand, to manage huge amounts of information in the form of unstructured content. To address this problem we propose the creation of scalable and flexible annotation processes that associate Web contents with semantic metadata, while still keeping the two spaces (content and metadata) decoupled (Section 4.1).

- **Usability**. Another important requirement in order to extend our ontology-based retrieval model to the Web environment is to provide users with an easy to use query

[19] http://www.powerset.com/

user interface (Section 4.2). This means not to require users to have previous knowledge of ontology-based query languages, or to navigate across complex forms to formulate their queries. To address this problem we propose the integration of a new query module that allows users to express their requirements using natural language.

Figure 3 shows the extensions performed over the previous framework in order to address the above mentioned challenges. Three main changes can be perceived in the architecture:

- The queries are not expressed using ontology-based query languages. Instead, queries are expressed in natural language as a compromise between expressivity and usability.
- The external resources for indexing and query processing are not a single ontology and KB, but online available SW information.
- In order to manage large amounts of semantic information during the query and annotation processes, a SW gateway is incorporated with the aims of gathering, storing and accessing the online distributed semantic information.

The overall retrieval process is illustrated in Figure 3, and consists of the following steps:

1. The system takes as input a user's natural language (NL) query. This query is processed by the query processing module, which has been replaced by an ontology-based Question Answering (QA) system, PowerAqua (López, Sabou, Uren & Motta, 2009). This component operates in a multi-ontology scenario where it translates the user terminology into the ontologies terminology. The integration of this QA system into our framework brings two clear benefits to our approach. First, the user interaction is eased by allowing natural language queries, improving the usability of the system. Second, the response is obtained from a large set of ontologies covering a potential unrestricted set of domains, therefore dealing with the heterogeneity limitation.
2. Once the pieces of relevant ontological knowledge have been returned as an answer to the user's query, the system performs a second step to retrieve and rank the documents containing this information. To do so, the document collection is automatically indexed in terms of the ontology concepts prior to the use of the system. The indexing module has been changed to integrate scalable and flexible annotation algorithms. These new indexing algorithms are able to deal with large document collections and large amounts of ontologies and KBs. Exploiting large amounts of metadata brings the advantage of retrieving Web documents without any potential domain restriction, therefore addressing the heterogeneity limitation.
3. The final output of the system consists of a set of ontology elements that answer the user's question and a complementary list of semantically ranked relevant documents.

[Figure 3 about here]

The details of how the main functionalities of our approach (document indexing, query processing, searching and ranking) have been adapted to exploit the information spaces defined by the SW and the (non-semantic) WWW are explained in the following subsections.

4.1 Indexing

In the proposed view of semantic search, it is assumed that the information available in standard Web pages (the document base) is indexed using the semantic knowledge found in the SW. A key step in achieving this aim lies on linking the semantic space to the unstructured content space by means of the explicit annotation of documents with semantic data. In such a dynamic and changing environment, annotation must be done in a flexible and scalable way. As we explain in the following sections, the solutions explored in this work do not require hardwiring the links between Web pages and semantic markup. On the contrary, these are created dynamically in such a way that the two information sources may remain decoupled.

Similarly to traditional IR techniques, which base their ranking algorithms on keyword weighting, our approach relies on measuring the relevance of each individual association between semantic concepts and Web documents. In this case, not just the retrieval process, but also the ranking of query answers can take advantage from the available semantic information.

Two different annotation methodologies are studied. The fist one uses Information Extraction methodologies in order to identify in the documents words or groups of words that can potentially represent semantic entities (classes, properties, instances or literals). The second one uses a more scalable approach based on statistical occurrences of semantic entities and their contextual semantic information. Both annotation procedures have been designed considering a set of common requirements:

- The semantic annotator identifies ontology entities (classes, properties, instances or literals) within the text documents, and generates the corresponding annotations. This is equivalent to a traditional IR indexing process where the indexing units are ontology entities (word senses) instead of plain keywords.
- The annotation processes carried out do not aim to populate ontologies, but to identify already available semantic knowledge within the documents. In this way, the semantic information and the documents remain decoupled.
- Differently to other large scale annotation frameworks, our system has been designed to support annotation in open domain environments. Any document can be associated or linked to any ontology without any predefined restriction. The exploitation of massive amounts of metadata and documents introduces scalability limitations. To address them, we propose the use of ontology indices, document indices, and non-embedded annotations:
    - Generation of ontology indices: We envision a scenario where the annotation module may need to interact with thousands of KBs structured in hundreds of ontologies. To successfully manage such amount of information on real time, the ontologies and KBs are analyzed and stored into

one or more inverted indices using Lucene[20]. This index structures are part of the SW gateway module explained in section 4.4.1.

- Generation of document indices. A massive amount of unstructured content is currently available on the Web. To successfully manage such amount of information on real time, Web documents are pre-processed and stored in one or more inverted indices using Lucene.
- Construction of the annotation database. In contrast to systems where annotations are embedded in the ontologies or documents, the proposed mechanism generates non-embedded annotations. These annotations are stored in a relational database, increasing the efficiency of the retrieval phase. For each annotation, an entry is generated in the database. This entry contains the identifiers of the corresponding semantic entity (word sense) and document, as well as a weight indicating the degree of relevance of the semantic entity within the document. Weights are automatically computed using different techniques for the two proposed annotation processes (see below).

In the following sections, we present the two implemented annotation processes. The first one analyzes textual documents using Natural Language Processing (NLP) techniques, extracts information from those documents, and maps it with the semantic information stored in the ontologies and KBs. The second one works in the opposite direction. It analyzes the semantic information stored in ontologies and KBs and, considering each ontology entity and its semantic context, attempts to identify the semantic entities within the textual documents to generate new annotations.

4.1.1 Annotation by NLP

Using Wraetlic NLP tools (Alfonseca, Moreno-Sandoval, Guirao & Ruiz-Casado, 2006), the annotation module analyzes the textual documents, removes stop words, and extracts relevant (simple and compound) terms, categorized according to their Part of Speech (PoS): nouns, verbs, adjectives, adverbs, pronouns, prepositions, etc. Then, terms are morphologically compared with the names of the semantic entities of the domain ontologies. The comparisons are done by using an ontology index created with Lucene (Section 4.4.1), and according to fuzzy metrics based on the Levenshtein distance (Levenshtein, 1966). For each term, if similarities above a certain threshold are found, the most similar semantic concepts are chosen and added as annotations of the document. After all annotations are created, a TF-IDF technique computes and assigns weights to them. Figure 4 shows a more detailed view of the annotation mechanism, which takes as input the HTML document to annotate, and the ontology indices, and returns as output new entries for the annotation database. The steps followed are:

1. The textual Web documents are parsed to erase meaningless (in terms of essential content to be conveyed) HTML tags.
2. The remaining text is analyzed by the Wraetlic tools to extract the PoS and the stem of each term.
3. The information provided by the linguistic analysis is used to filter the less meaningful terms or stop words (determinants, prepositions, etc.), and to identify those sets of terms that can operate as individual information units.
4. The filtered terms are searched in the ontology indices, obtaining the subset of semantic entities to annotate.
5. The annotations are weighted according to the semantic entity frequencies within individual documents and the whole collection.
6. The annotations are added to a relational database.

[Figure 4 about here]

Enhancing the accuracy of annotations

The use of a potentially unlimited number of domain ontologies ad KBs increases the uncertainty of the annotations, since more morphological similar concepts (with divergent meanings) can be found. To address this limitation, we propose to exploit the PoS information provided by Wraetlic NLP tools in order to identify and discard those words that typically do not provide significant semantic information. Moreover, the approach attempts to group sets of words that can operate as individual semantic information units. The empirically identified patterns are the following:

- Noun + noun. E.g., “tea cup”.
- Proper noun + proper noun. E.g., “San Francisco”.
- Proper noun + proper noun + proper noun. E.g., “Federico García Lorca”.
- Abbreviation + proper noun + proper noun. E.g., “F. García Lorca”.
- Abbreviation + abbreviation + proper noun. E.g., “F. G. Lorca”.
- Participle + preposition. E.g., “located in”, “stored in”.
- Modal verb + participle + preposition. E.g., “is composed by”, “is generated with”.

Weighting annotations

As in the base model (section 3.1), annotation weights are computed automatically by an adaptation of the TF-IDF algorithm, based on the frequency of the occurrences of each semantic entity within the document. The number of occurrences of a semantic entity in a document is primarily defined as the number of times any of its associate keywords appears in the document text. In our first experiments, we observed that quite a number of occurrences were missed in practice, since the algorithm was not considering pronouns as semantic entity occurrences. To mitigate this limitation, a modification of the algorithm has been introduced to count pronoun occurrences in the scope of a sentence if a noun-based semantic entity has been previously identified.

The Wraetlic tools use the PoS tags of the Penn Treebank corpus[21]. These PoS classification distinguishes between four different types of nouns (NN Noun singular or mass, NNS Noun plural, NNP Proper noun singular and NNPS Proper noun plural) and four different types of pronouns (PRP Personal pronoun, PRP$ Possessive pronoun, WP Wh-pronoun and WP$ Possessive Wh-pronoun). Wh-pronouns

[20] http://lucene.apache.org/java/docs/index.html

[21] The Penn Treebank Project, http://www.cis.upenn.edu/~treebank/

are not used for counting additional occurrences. In the rest of the cases, the following simple rules are followed:

- A PRP$ refers to the previous NNP or NNPS if it exists and their numbers (singular, plural) agree.
- A PRP refers to the previous NNP or NNPS if it exists and their numbers (singular, plural) agree, except in the case of the pronoun "it" ("it" always refers to the previous NN).

Note that these rules do not cover all the cases and, therefore, not all pronoun occurrences are taken into account. However, we observed that, following these simple rules, most of the identified pronoun occurrences were correctly associated with their corresponding semantic entity. As future research work, we plan to exploit current state of the art techniques in coreference resolution (Ponzetto & Poesio, 2009) to detect not only pronouns (*he*), but also nominal (*president*) and proper (*Barak Obama*) mention types. While this modification in the weighting algorithm does not help to increase the correctness of the annotations, or to obtain new ones, it enhances the accuracy of the annotation weights that will be later used in the ranking process.

4.1.2 Annotation based on contextual semantic information

In the NLP based annotation mechanism, the documents are analyzed to filter the terms that have to be searched in the semantic entity index. Here, instead, the semantic entities are those analyzed and searched in the document index, a standard keyword-based index generated prior to the annotation process. Inverting the direction of the annotation process from semantic entities to documents, provides two important advantages: on the one hand, the semantic information stored in the ontologies and KBs can be used as background knowledge to improve the accuracy of the annotations; on the other hand, the computational cost decreases because the textual documents have been indexed in advance. This new annotation schema constitutes a more scalable and widely applicable approach because it can potentially use any keyword-based document index. The overall annotation process is shown in Figure 5, and consists of the following steps to be performed for every semantic entity in every ontology:

1. Load the information of a semantic entity, that is, extracting the textual representation of the selected entity. Each entity may have one or more textual representations in the ontology. For instance, the individual entity describing the football player Maradona can be named as "Maradona", "Diego Armando Maradona", "Pelusa", etc. Here, we assume that such lexical variants are present in the ontology as multiple values of the local name or *rdfs:label* property of the entity.
2. Find the set of potential documents to annotate. The textual representations of the entity are then searched in the document index using standard searching and ranking techniques. The retrieved documents simply contain the textual representation of the entity, which does not necessarily imply that they contain its meaning. The disambiguation process is performed in the subsequent steps by exploiting the context of the entity in the ontology.
3. Extract the semantic context of the entity. The meaning of an entity is determined by the set of concepts it is related to in the domain ontology. To ensure that the entity annotates the appropriate set of documents, the ontological relations are exploited to extract its semantic context, that is, the set of entities directly linked in the ontology by explicit relations. Following the example described in Figure 5, the semantic context of the entity Maradona in the ontology is formed by the entities "football player" and "Argentina".
4. Find the set of contextualized documents. The textual representations of entities belonging to the semantic context are then searched in the document index to extract the set of contextualized documents. In the example, the textual representations "Argentina" and "football player" are used to extract the set of contextualized documents.
5. Select the final list of documents to annotate. We compute the intersection between the set of documents containing any textual representation of the entity (extracted in step 2), and the set of documents containing any textual representation of its semantic context (extracted in step 4). Documents in this intersection are not just likely to contain the corresponding entity, but also the contextual meaning of the entity in the ontology. In our example, this documents do not only contain any of the textual representations of the concept "Maradona", but also at least one of the textual representations of its semantic context ("football player", "Argentina")
6. Create the annotations. A new entry or annotation is created for each document in the previous intersection. The annotation is assigned a weight indicating the degree of relevance of the entity within the document. The algorithm to calculate this annotation weight is explained below.

[Figure 5 about here]

Enhancing the accuracy of annotations

As described previously, to reduce the ambiguity of annotations, the context of the semantic entities is taken as background information. The context of a semantic entity is defined as the set of entities directly linked to it in the ontologies by explicit relations. Using this context, we are able to annotate entities with documents that contain the ontological meaning of the semantic entity. We have empirically observed that this technique brings a considerable precision gain, but at the expense of losing an important number of annotations. A potential cause of this problem is the low density of relations supported in the SW ontologies (d'Aquin, Gridinoc, Sabou, Angeletou & Motta, 2007). There are cases where the ontologies do not have enough contextual information to identify the meaning of the entity in the document, and, therefore, the annotation is not created. This trade-off between the quality and the quantity of annotations is further investigated in Section 5.3.2.

Weighting annotations

In both the base model and the NLP-based annotation schemas, annotations weights are computed automatically by an adaptation of the TF-IDF algorithm based on the frequency of the occurrences of each semantic entity within

the document (Castells, Fernández & Vallet, 2007). In the contextual annotation approach, the annotation weights are computed as follows:

- A fusion technique, described in (Fernández, Vallet & Castells, 2006), is applied to the ranked lists of documents obtained from steps 2 and 4 to produce a ranked list *S* of documents, candidates to be annotated, and a ranked list *C* of contextualized documents for semantically related entities.
- A document *d* appearing in both lists *S* and *C* is selected for annotation by step 5, and is assigned a weight $\lambda \cdot S_d + (1-\lambda) \cdot C_d$, where λ is a constant used control the influence of the semantic contextualization, S_d is the score of document d in the ranked list *S*, and C_d is the score of document d in the ranked list *C*. A value of $\lambda = 0.6$ was empirically found to work well in our experiments.

This annotation weighting approach is less sensitive to potential changes in the ontologies and KBs. When a new semantic entity is added or modified, it is only necessary to recompute its annotations, and the annotations of the semantic entities directly linked to it in the ontologies and KBs. However, it presents one main trade-off: the use of document ranking scores to compute the annotation weights introduces a loss of accuracy with respect to our previous weighting techniques.

4.2 Query processing

As already mentioned, most semantic search systems suffer from one of two following limitations when attempting to enhance the conceptual representation of user needs beyond plain keywords: a) usability limitations, where users are expected to use formal query languages to express their requirements and, b) heterogeneity limitations, where a predefined (usually small) set of ontologies is used as the target data set.

Aiming to overcome these limitations, we use an ontology-based QA system, PowerAqua (López, Sabou, Uren & Motta, 2009), as the query processing module. PowerAqua is able to answer queries by locating and integrating information, which can be massively distributed across heterogeneous semantic resources. To do so, PowerAqua initially uses syntactic techniques to identify those semantic resources which may be relevant to the user query. In many cases this initial syntactic match will generate several possible candidates (i.e., semantic entities), which may provide potential alternative interpretations for a query term. Hence, to address this problem, PowerAqua builds on techniques developed in the Word Sense Disambiguation community, to disambiguate between different possible interpretations of the same query term within an ontology or across several ones. In particular, PowerAqua makes use of the context of the user query, the context surrounding candidate entities in their ontologies, and the background knowledge provided by WordNet to determine the most likely interpretation or a user query as whole and the individual terms in the query.

For instance, let's consider the query "*Rock musicians in Britain*". Firstly, PowerAqua parses the query and translates it into a set of *linguistic triples*: <*rock, ?, musicians*>, <*musicians, ? , Britain*>. In a second step, PowerAqua then searches for approximate syntactic matches of these linguistic triples in the ontologies, using not just the terms in the triples, but also lexically related words obtained from WordNet (synonyms, hypernyms and hyponyms). In this case, the term *rock* belongs to different WordNet synsets, and is therefore associated to different meanings, *stone* and *music genre*. However, by matching the linguistic triples to candidate answer triples in the relevant ontologies, and by determining the most likely WordNet senses for the potentially ambiguous entities in the ontologies, in this example, PowerAqua will quickly find that no ontologies relate musicians to rock interpreted as stone, while many ontologies relate musicians to rock, interpreted as a music genre. Hence, PowerAqua can discard the sense of rock as a stone. In those cases, where it is not possible to quickly determine the correct sense for a query term, i.e., alternative interpretations are covered by one or multiple ontologies, then PowerAqua will produce a ranking of the different interpretations according to their *popularity* – i.e., how many ontologies in the result set contain each particular interpretation.

This approach has been evaluated empirically and has been shown to perform well with our evaluation datasets. More details on both the algorithm and the evaluation studies can be found in (López, Sabou & Motta, 2006; Gracia et al., 2007).

4.3 Searching and ranking

The semantic document retrieval and ranking approach presented here is the same as the one in our initial design (Section 3.2), except for the way in which the query vector is constructed. As explained earlier, the document retrieval and ranking algorithm is based on an adaptation of the traditional vector space IR model, where documents and queries are represented as weighted vectors. The query vector components represent the importance of each semantic entity in the information need expressed by the user, while the document vector components represent the relevance of each semantic entity within the document.

The construction of the document vector remains from our previous model, but the construction of the query vector has been adapted to manage the degree of uncertainty of the answers retrieved by PowerAqua. Note that, in the base model, the input was a formal SPARQL query. This query was executed against the KB returning as answer a list of instance tuples in a purely Boolean step (i.e., based on an exact matching). Using PowerAqua as query processing module introduces a degree of uncertainty in the retrieved answers: firstly, because it searches for approximate syntactic matches in order to find the ontologies that can potentially answer the user's query; secondly, because it has to disambiguate the sense of the identified entities using as background knowledge the available semantic information; and finally, because it constructs one or more generic patterns to match the tuples from different ontologies.

To compute the degree of uncertainty of the retrieved answers and, as a simple approach to weight the elements of the query vector, a query weighting measure has been introduced into the query module. This measure is based on the number of semantic entities retrieved for each detected query condition. For example, if the user asks for "symptoms and treatments of Parkinson disease", PowerAqua is able to retrieve as answer a set of individual symptoms and a set of individual treatments. Considering that SE_{ci} is the set of semantic entities retrieved for the query condition *i*, the weight of each retrieved semantic entity in

the query vector is computed as 1 / $|SE_{ci}|$. The intuition behind this measure is that those query variables for which fewer ontology entities have been retrieved are more likely to be representative of the user's information needs, and therefore they should be considered more important.

4.4 Providing fast access to SW information: a SW Gateway

This section focuses on the work carried out towards the generation of a SW gateway that collects, analyzes and gives access to online available semantic content, enabling the experimentation of the proposed retrieval algorithms on large amounts of semantic content. A SW gateway should accomplish three main goals:

- Collect the available semantic content from the Web.
- Implement efficient storage facilities to access the data.
- Implement ontology evaluation and selection algorithms to retrieve the most appropriate semantic information considering the user or application needs.

One of the most popular SW gateways currently available is **Swoogle** (Ding, Finin, Joshi, Pan & Cost, 2004). This system claims to have indexed around ten thousand ontologies, which is a significant coverage of the SW data. However, the selection algorithms that this tool provides to users and applications are based on traditional IR methodologies, like the well known Page-Rank algorithm (Page, Brin, Motwani & Winograd, 1999). Thus, the ontology selection algorithms do not take into account semantic data quality measures such as lexical vocabulary, relations, consistency, correctness, etc.

Another very popular SW Gateway is **Watson**[22] (d'Aquin et al., 2007). It combines the capabilities of Swoogle to crawl and search SW data with novel techniques to analyze the quality of content.

For the purpose of having control over our experimental environment, and making our evaluation reproducible, we developed our own SW gateway, **WebCORE** (Fernández, Cantador & Castells, 2006; Cantador, Fernández & Castells, 2007), focusing our attention in the last two requirements and avoiding the cost of constructing a SW crawler. The collection of semantic content has been done manually from several public ontology repositories, and contains around 2GB of metadata. The designed structures to store and access the semantic content are explained below. These structures are used by the annotation modules (Section 4.1), and by the introduced natural language query processing module, PowerAqua. Its algorithms for ontology selection and evaluation are described in (Fernández, Cantador & Castells, 2006; Cantador, Fernández & Castells, 2007), but are not used as part of this work.

4.4.1 Ontology indexing module

To efficiently access large amounts of SW content, WebCORE pre-processes and stores the gathered information in several inverted indices. Two kinds of indices are created, the **lexical ontology index**, which associates each semantic entity (class, property, instance or literal) with a set of terms or lexical representations, and, the **taxonomical ontology index**, which associates each semantic entity with its direct subclasses and superclasses.

[22] http://watson.kmi.open.ac.uk/WatsonWUI/

The lexical ontology-index generation is achieved by a concept-keyword extraction mechanism over the semantic entities. The keywords associated to each concept are extracted from the entity local name (which is part of its URI), the standard ontology meta property *rdfs:label*, and optionally, from any other ontology property.

An example of the generated inverted index is shown in Table 3, where each keyword is associated to one or several semantic entities from different ontologies. The semantic entities are uniquely identified within the system by the identifier of the ontology they belong to, their URIs, their type (class, property, individual or literal), and their set of associated terms obtained after the concept-keyword extraction phase.

These indices are useful to identify, in a first step, the set of potential semantic entities (over the whole gathered SW content) that can be associated to a set of pre-defined terms describing a user query, a document, or any other application need.

[Table 3 about here]

To search the set of semantic entities associated to a specific term in the indices, we make use of the search capabilities of Lucene, and the term relations obtained with WordNet. Lucene allows performing three different kinds of searches within the lexical ontology index:

- *Exact search*: the index must contain the exact searched term to retrieve an answer.
- *Fuzzy search*: the keywords stored in the index must be "similar" to the searched term. The similarity is computed using the Levenshtein distance (Levenshtein, 1966), and considering an established prefix that represents the number of letters that must be equal at the beginning of both words.
- *Spell search*: the searched term might contain some spelling mistakes. In this case, Lucene provides some suggestions of additional terms. For these cases, the system uses the first suggestion in order to perform a new search within the index.

WordNet allows extending the searched terms with three main types of relationships: synonyms, hypernyms and hyponyms. Searching for related terms increases the chances of finding a matching within the index.

A second index level is generated to store taxonomical information. In this way, the ontology entities are also associated with its main superclasses and subclasses. An example of this indexing structure can be shown in Table 4.

[Table 4 about here]

The lexical and taxonomical indices increase the mapping speed of semantic entities, allowing the management in real time of the distributed semantic information. For those cases in which the system requires more information than the one stored in the indices, the SW gateway provides a multi-ontology accessing module that allows managing several ontologies at a time within the application.

4.4.2 Multi-ontology access module

Providing universal access to multiple ontologies from different applications presents two main difficulties: accessing the semantic content in a common way for all the applications, and generating appropriate multi-ontology

management modules to administer several ontologies at a time. Three main problems should be addressed in order to access the semantic content in a common way for all the applications:

- Ontologies are expressed in different languages (RDF, OWL, DAML, etc.)
- Ontologies can be stored in different types of repositories (databases, text files, URLs, etc.)
- Ontology frameworks implement different APIs to access ontologies (Sesame[23], Jena[24], etc.)

To address the first problem, we have developed a common API to access all the distributed semantic content. Figure 6 shows the architectural design and the set of layers involved in the semantic content accessing process.

[Figure 6 about here]

At a first level, an OntologyPlugin API defines a common set of functionalities to query ontologies and KBs independently of their language, type of storage, and location. In a second layer, two implementations of the above API are provided for two of the most popular SW frameworks: Sesame and Jena. Different extensions of the implementations are done for these frameworks to encapsulate the different ontology languages, and types of storage. These implementations are done using the APIs and the query languages available for the different SW frameworks, which are the ones that directly access to the SW graph of information.

Ontologies are added to WebCore by providing the following information: the ontology identifier, its language, its corresponding framework, and its location. An example of such information is shown in Figure 7. The SW gateway manages this information for the ontologies that have been previously gathered from the SW. However, any external application can provide information about a new ontology, so that the SW gateway can analyze, and store or access it at run time.

[Figure 7 about here]

To manage several ontologies at a time in the application, the SW gateway provides an additional API to encapsulate a cache based memory of ontologyPlugins. Internally, this cache structure is managed as a Hash table of OntologyPlugins, where each ontologyPlugin is associated to the ontology identifier. A graphical view of the MultiOntologyPlugin structure is shown in Figure 8.

[Figure 8 about here]

With the described architecture, the SW gateway provides large-scale storage and accessing capabilities. It provides the necessary structures to manage multiple ontologies and KBs at the same time, and provides a common API to access the semantic content independently of the ontology language, storage type, and location.

5 EVALUATION

In contrast to the IR community, where evaluation using standardized techniques, such as those used for the annual TREC competitions, has been common for decades, the SW community is still a long way from defining standard evaluation benchmarks to judge the quality of semantic search methods. Current approaches for SW technology evaluation are based on user-centred methods (Sure & Iosif, 2002; McCool, Cowell & Thurman, 2005; Todorov & Schandl, 2008), and therefore tend to be high-cost, non-scalable and difficult to repeat, especially at Web scale.

A systematic and rigorous evaluation thus involves addressing a benchmark building task. We require a text collection, a set of queries and the corresponding document judgments, ontologies covering the query topics, and KBs populating such ontologies, preferably using an independent source, which is independent from the text collection.

5.1 Evaluation benchmark

The constructed benchmark is composed by: a) the TREC WT10G document collection (Bailey, Craswell & Hawking, 2003), b) 20 queries selected and adapted from the TREC9 and TREC2001 competitions[25], with their corresponding judgements, c) 40 public ontologies covering a subset of TREC domains (these comprises 370 files, and around 400MB of RDF, OWL and DAML), plus 100 additional repositories (2GB of RDF and OWL) stored, indexed and accessed towards the SW gateway integrated into the system, and d) the public available KBs associated with these ontologies, plus some metadata generated from a external data source, Wikipedia. The detailed motivation, principles, and steps for the construction of this evaluation benchmark, as a step forward in the standardization of a semantic search evaluation methodology, are given in (Fernández et al., 2009).

5.2 Experimental set up

The proposed experimental setup involves the comparison of four different systems: three traditional keyword-based systems (Lucene, the best TREC automatic, and the best TREC manual, both of them from TREC9 and TREC2001 competitions), and our semantic search engine. The difference between the best TREC automatic (*short runs*) and manual (*notshort runs*) is based on how they process the query (Hawking, 2000). While automatic search approaches take the query as it comes from user logs, TREC manual approaches modify the query with additional information before processing it. Note that the results by TREC manual are reported but are not included as part of the discussion because, as mentioned in (Hawking, 2000), "only short runs are representative or real Web search. Non-short runs increase the number of known relevant documents and give an idea of what level of performance may be possible on each task".

5.3 Results

Tables 5 and 6 show the results of the evaluation using the 20 TREC topics and two standard IR evaluation metrics: Mean Average Precision (MAP) and Precision at 10 (P@10) for each of the evaluated search approaches. The first metric captures the overall performance of the system in terms of precision, recall and ranking. The second one relates to the accuracy of the top-10 results, which are generally the ones most often explored by search users.

[23] OpenRDF, http://www.openrdf.org/

[24] Jena Semantic Web Framework, http://jena.sourceforge.net/

[25] http://technologies.kmi.open.ac.uk/poweraqua/trec-evaluation.html

Values in bold correspond to the best results for the corresponding topic and metric, excluding the Best TREC manual approach, which outperforms the others significantly by both metrics, likely because of manual modifications to the query. Note that, for this experiment, the semantic retrieval approach uses the annotation process based on contextual semantic information described in section 4.1.2.

[Table 5 about here]

[Table 6 about here]

As shown in Table 6, **by P@10, the semantic retrieval outperforms the other two approaches**, providing the highest quality for 55% of the queries, and being only outperformed by both Lucene and TREC semantic for one query (511). Semantic retrieval provides better results than Lucene for 60% of the queries, and equal results for another 20%. Compared to the best TREC automatic engine, our approach performs better for 65% of the queries, and produces comparable results for 5%. Indeed, the highest average value for this metric is obtained by semantic search.

The results by MAP, in Table 5, show that there is no clear winner. While the average rating for Best TREC automatic is higher than that for semantic search, the latter outperforms TREC automatic in 50% of the queries, and Lucene in 75% of the cases.

We hypothesize that the quality of the results retrieved by semantic retrieval, and its measurement with MAP may be disadvantaged by the following two factors:

- More than half of the documents retrieved by the semantic retrieval approach lack relevance judgments in the TREC collection. Therefore, the used metrics marked them as irrelevant, when, in fact, some of them can be considered relevant. In Section 5.3.1, we study the impact of this effect by manually evaluating some results to analyze how the semantic retrieval approach would perform if all the documents had been evaluated.
- The annotation process used for the semantic retrieval approach is very restrictive (see section 4.1.2). In order to increase the annotation accuracy, an annotation is generated when a document contains not just a concept, but also its semantic context. If the concept appears in the document with a semantic context not covered by its ontology, the annotation is not generated. Thus, the process discards potentially correct annotations. The impact of this effect is studied in section 5.3.2, which compares the performance of the context-based annotation model with the NLP based annotation model (see section 4.1.1).

The aforementioned sections also explain why these factors affect the MAP measurements much more than the P@10 measurements.

Three additional relevant conclusions can be drawn from the evaluation:

- **For some queries for which the keyword search (Lucene) approach finds no relevant documents, the semantic search does**. This is the case of queries 457 (*Chevrolet trucks*), 523 (*facts about the five main clouds*) and 524 (*how to erase scar?*). When the same reality is expressed in document and queries using a different terminology, semantic search approaches, as opposed to traditional keyword-based models, are able to detect it and provide valuable answers.
- The queries in which the semantic retrieval did not outperform the keyword baseline seem to be those where the semantic information obtained by the query processing module was scarce. One of such queries would be 467 (*Show me all information about dachshund dog breeders*). **However, the keyword-based baseline only rarely provides significantly better results than semantic search**. In addition, it is important to highlight that, when there is a lack of semantic information to answer the users' request, the system "gracefully degrades" to a keyword based system and as a result its performance is still comparable to traditional keyword based approaches.
- As already pointed out, the effect of complex queries (in terms of relationships) was not evaluated, because TREC search evaluation topics are written for keyword-based search engines, and do not consider this type of query expressivity. Based on prior experiments outside TREC (Castells, Vallet, Fernández, 2007) it is fair to assume that for more complex queries, involving several related information needs or potentially ambiguous meanings, **the performance of semantic search is likely to improve significantly** relative to the baselines.

5.3.1 Studying the impact of retrieved non-evaluated documents

Given a TREC topic and a document, there is one of three possibilities:

- The document is judged as a relevant result.
- The document is judged as an irrelevant result.
- The document has not been judged in the TREC collection. If semantic search retrieves it, our metrics take it as irrelevant.

As shown in Table 7, **only 44% of the results returned by semantic retrieval had been previously evaluated** in the TREC collection. The non judged documents, 66% of the whole set, are therefore considered irrelevant beforehand, although some of these results may be relevant. Therefore, the performance of semantic retrieval may be better than reported.

[Table 7 about here]

Figure 9 shows the probability of a result returned by the semantic retrieval approach to be evaluated as function of its position. Results in the first positions have a very high probability. In other words, the first results returned by the semantic retrieval approach are very likely to have been returned by at least one of the TREC search engines as well. This explains why unevaluated results are a significant issue for MAP but not for P@10.

[Figure 9 about here]

We now focus on how the lack of evaluations for documents retrieved by semantic search does affect the results by the MAP metrics. A legitimate question is whether the unevaluated results are actually relevant. Indeed, a result is no evaluated if it was not returned by any of the search

engines in TREC, which one may expect to imply that it has a low probability of being relevant.

To provide a partial answer to this question we performed a small evaluation of the first 10 non evaluated results returned for every query: a total number of 200 documents. 89% of these results occur in the first 100 positions for their respective queries. The first 10 were picked because these are the most likely to be seen by the user, and also because, occurring first on the query, have a larger impact on the MAP measurements. Note that these additional judgments had not been used to conduct the global evaluation, shown in Tables 5 and 6, but just as a posteriori study of the fact that semantic search is finding additional documents that none of the TREC participants had retrieved. Note also that the assessor performing this evaluation is a different person than the original TREC assessor. This may lead to an inter-assessor consistency problem in which the new assessor may be interpreting relevance in a stricter or a looser way than the original one.

The results of the evaluation are given in Table 8. For each query, we show the percentage of documents judged as relevant that were not evaluated in TREC, and the average, minimum and maximum ranking positions of those documents.

[Table 8 about here]

A significant portion, 31.5%, of the additional judged documents turned out to be relevant. Clearly, this cannot be generalized to all the non evaluated results returned by the semantic search approach: as one moves towards the bottom, the probability of a result being relevant decreases, as shown by Figure 10. This figure is based only on the TREC evaluations, treating non evaluated (by TREC) results as irrelevant, so the actual probability is slightly higher. The figure shows that the probability of being relevant drops around the first 100 results, and then varies very little. Although this percentage cannot be generalized, it supports our hypothesis that the MAP value of the semantic search approach puts this at a disadvantage with respect to the MAP value obtained by TREC approaches.

[Figure 10 about here]

We analyzed the queries for which at least 50% of the top-10 documents retrieved and not evaluated by TREC were considered relevant in our evaluation. The analysis shows that, in most of these cases, semantic search was obtaining new relevant documents when the query involved a class-instance relationship in the ontologies. Examples of such queries are: "*symptoms and treatments of Parkinson disease*" or "*movies or TV programs where Jenifer Anniston appears"*. This effect indicates that, semantic search obtains better recall when querying for class instances.

Most of the results evaluated and listed in Table 8, even those considered as irrelevant, contain information related to the query. For example, for topic 451, "*What is a Bengals cat*", although documents about Bengal cats were not retrieved, most of the results were about other types of cats. For topic 457, the results focused on specifications of Chevrolet cars instead of Chevrolet trucks. This potential "recommendation" characteristic of semantic search could even have a positive impact on the user's satisfaction, but this should be studied more carefully before definitive conclusions can be drawn.

5.3.2 Annotation quality vs. quantity tradeoffs

As Table 8 shows, many relevant documents retrieved by the TREC search engines were not retrieved by the semantic search approach.

We hypothesize that the restrictions in the annotation process may have some influence here. Note that annotations are only generated if the ontological context of the entity is found within the documents (see section 4.1.2). This loss of potential correct annotations is a price to be paid for the increase in accuracy.

We decided to run a small-scale test with a variation of the annotation process based on NLP methodologies (this annotation method is described in section 4.1.1). Although this new annotation method is less restrictive, given the fact that it relaxes the conditions to generate a new annotation, and its weighting algorithm generates more accurate weights (it is based on an adaptation of TF-IDF over semantic entity frequencies), it is important to stress that this annotation model is considerably less scalable. Note that, even though the annotation is an off-line process, the context-based annotation model is based on traditional keyword document indices, and could thus take advantage of the structures used by large commercial search engines, such as Google or Yahoo!.

With the aim to compare the effect of the two different annotation processes, four topics were selected based on the total percentage of retrieved non evaluated documents: 484 (13%), 452 (31.3%), 465 (38.5%), and 476 (50.6%).

The results of the analysis are shown in Tables 9 and 10. For each query, these tables include:

- The old value of the metric for the semantic search approach.
- The new value of the metric for the semantic search approach.
- The value of the metric for the Best TREC approach.

[Table 9 about here]

[Table 10 about here]

As shown in the tables, the quality of results increases significantly with the new annotation model. On average, by the MAP metric, the new model performs 1.76 times better than with the previous annotation method. What is more, the quality of the first results, measured by P@10, did not diminish: in fact, it went up (albeit marginally). This is due to the highest accuracy of the weighting annotation algorithm used for this annotation model, which is based on TF-IDF algorithms.

Besides the selected algorithms, two important factors also affect the quality and quantity of annotations, and therefore, the effectiveness of the semantic retrieval: a) the volume and domain coverage of the publicly available semantic information and, b) how this information is associated or "annotated" with the current World Wide Web content.

Regarding the first factor, major advances have been made in the last few years, creating rich semantic resources, such as DBPedia[26] and Freebase, and opening up large datasets previously hidden under backend databases, like the ones released by the data.gov[27] initiative.

[26] DBPedia: http://dbpedia.org
[27] http://data.gov.uk/

Regarding the second factor, initiatives like Google Rich Snippets and Yahoo! Search Monkey are encouraging publishers to annotate their own Web content. In a recent publication[28], Google declared that currently 5% of the Web pages have some semantic markup, but they expect to raise soon this number up to 50%. Additionally, for the cases in which publishers do not introduce this information, current state of the art techniques (Zaragoza et al., 2008) have shown the feasibility of performing rapid analysis of Web content to generate large-scale annotation.

Considering all these advances seems reasonable to assume that an important percentage of Web content will be soon annotated with semantic information with an acceptable coverage. In addition, for those cases in which the annotations are scarce, our semantic retrieval model is combined with a traditional keyword-based retrieval technique to maintain an appropriated level of recall.

5.3.3 Run time performance evaluation

Table 11 shows the performance of the semantic search system in terms of query response time, showing a maximum time of 18.32 seconds for query 523, "*facts about the five main clouds?*", a minimum time of 1.63 seconds for query 491 "*Japanese Wave*" and an average of 5.37 seconds per query.

[Table 11 about here]

The experiments have been conducted on a server with the following characteristics: 3GHz Intel Pentium Dual Core, 8GB RAM and 150 GB hard disk.

As described by the results, the query time, even though reasonable, is still too high. Most of this execution time corresponds to the query processing phase (Section 4.2). Although PowerAqua uses the generated ontology indices (Section 4.4.1) and multi-ontology access module (Section 4.4.2) to filter the ontologies that may partially cover the user's request, extracting the answer from these ontologies may require the execution of several ontology-based queries over the corresponding triple stores where the ontologies are saved. To conduct these experiments, Sesame 1.x[29] has been used as the main triple store. However, current initiatives to evaluate and compare the performance of triple store systems at a large scale[30] have shown that, while Sesame is faster for smaller repositories, Virtuoso[31] outperforms it in query time for large metadata volumes. Thus, our main strategy to improve the performance is to replace Sesame 1.x by Virtuoso as the main triple store. Parallel computing, and more specifically Hadoop[32], has been also considered to reduce the total query processing time, dividing the tasks carried out by PowerAqua, and minimizing its response time to the performance of the slowest ontology-based query executed over the selected triple store.

[28] Google Semantic Technologies invited talk, San Francisco, 2010: http://www.readwriteweb.com/archives/google_semantic_web_push_rich_snippets_usage_grow.php

[29] http://www.openrdf.org/documentation.jsp

[30] http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/results/index.html

[31] http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/

[32] http://hadoop.apache.org/

6 CONCLUSIONS AND FUTURE WORK

The main goal of this work is to attempt to bridge the gap between the IR and the SW communities in the understanding and realization of semantic search. In order to leverage the best features towards semantic search from both fields, and with the ultimate goal of improving the retrieval performance of traditional keyword-based search, this work proposes the generation of a novel semantic search model that integrates and exploits highly formalized semantic knowledge in the form of ontologies and KBs within traditional IR ranking models.

As a further extension of this research line, we have investigated the practical feasibility of applying semantic search models to the Web environment. Several problems, among which we may highlight the size and heterogeneity of the content or the need for simple ways of interaction with users, keep this line of research open to further improvements. Our contributions here are based on providing potential solutions to the above mentioned problems, taking a step towards the advancements of semantic search models within large scale and heterogeneous environments, such as the Web. This goal has been achieved by:

- The integration of an external NL query processing module, PowerAqua (López, Sabou, Uren & Motta, 2009). This integration aims to solve the problem of usability, allowing the user to express her requirements in natural language, and the problem of heterogeneity, exploiting PowerAqua's ability to answer queries using large amounts of heterogeneous semantic content.
- The implementation of flexible and scalable annotation algorithms that generate annotations between large amounts of documents and semantic metadata, while maintaining both information sources decoupled.
- The use of a SW gateway that provides fast access for applications to SW content.

The evaluation of this model has been done using a large-scale evaluation benchmark based on an adaptation of the evaluation benchmarks used for the TREC Web track competition (Fernández et al., 2009). The generated benchmark allows the comparison between semantic search systems and traditional keyword-based search approaches.

As a side effect of an ontology-based approach, we also investigate the problem of knowledge incompleteness. This problem refers to the need of retrieving accurate results when the semantic information is not available or incomplete. To address this problem we propose the combination of rankings coming from ontology-based and keyword-based search results. This combination is based on a score normalization algorithm (Fernández, Vallet & Castells, 2006), that undoes potential biases in the distribution of the scores.

As general conclusions of this work we would like to highlight that:

- Semantic retrieval approaches can integrate and take advantage of SW and IR models and technologies to provide better search capabilities, thus achieving a qualitative improvement over keyword-based retrieval by means of the introduction and exploitation of fine-grained domain ontologies.

- The application of semantic retrieval models to the Web, and more specifically the integration of ontologies as key-enablers to improve search in this environment, remains an open problem. Challenges and limitations such as the size and heterogeneity of the Web, the scarceness of the semantic knowledge, the usability constraints, or the lack of formal evaluation benchmarks, can be pointed out as some of the main reasons for the slow application of the semantic retrieval paradigm at Web scale.

As reflected in this work, the topic of semantic search is very broad, and involves many different aspects that can be addressed as future research lines including unsolved limitations, possible courses of action to address them, and potential future research challenges. Among the most relevant future lines of work we would like to highlight:

- The exploitation of richer, in terms of amount of semantic information, SW gateways, e.g., Watson (d'Aquin, Baldassarre, Gridinoc, Angeletou, Sabou & Motta, 2007).
- The analysis of subsets of queries for which semantic search algorithms generally perform better than traditional keyword-based approaches (like those ones involving potential different meanings or complex information needs).
- The analysis of the effect of semantic coverage, i.e., the comparison of results obtained with rich custom-made generated ontologies and KBs against the results obtained reusing public online available ontologies.
- The study and development of novel evaluation benchmarks, which consider not just basic IR performance measures, such as precision and recall, but which also measure the new features associated with semantic search paradigms (different ways of interaction, richer user interfaces, structured answers, etc.).

To conclude, in this paper we have presented a comprehensive semantic search model which, extends the classic IR model, addresses the challenges of the massive and heterogeneous Web environment, and integrates the benefits of both keyword and semantic-based search. The evaluation results have shown that the semantic search model outperforms the best TREC automatic (*short run*) system, thus demonstrating the value of the approach. Future research lines will focus on exploiting the richer amounts of semantic information, like the ones emerged under the LOD initiative, with the final aim to provide better levels of information coverage when answering users' information needs.

7 REFERENCES

Alfonseca, E., Moreno-Sandoval, A., Guirao, J. M., & Ruiz-Casado, M. (2006). The Wraetlic NLP Suite. *In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy.

Baeza Yates, R., & Ribeiro Neto, B. (1999). Modern Information Retrieval. Harlow, UK: Addison-Wesley.

Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for web. *Information Processing and Managment*, 39(6), pp. 853-871.

Bernstein, A., & Kaufmann, E. (2006). Gino - a guided input natural language ontology editor. *In Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pp. 144-157. Athens, GA, USA.

Cantador, I., Fernández, M., & Castells, P. (2007). Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments. *In Proceedings of the International Workshop on Social and Collaborative Construction of Structured Knowledge, at the 16th International World Wide Web Conference (WWW 2007)*. Banff, Canada.

Castells, P., Fernández, M., & Vallet, D. (2007). An Adaptation of the Vector space Model for Ontology-based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2), pp. 261-272.

Chirita, P. A., Gavriloaie, R., Ghita, S., Nejdl, W., & Paiu, R. (2005). Activity based metadata for semantic desktop search. *In Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, 439-454. Heraklion, Greece.

Cimiano, P., Haase, P., & Heizmann, J. (2007). Porting Natural Language Interfaces between Domains - An Experimental User Study with the ORAKEL System. *In Proceedings of the International Conference on Intelligent User Interfaces (IUI 2007)*, pp. 180-189.

Cohen, P., & Kjeldsen, R. (1987). Information Retrieval by constrained spreading activation on Sematic Networks. *Information Processing & Management,* 23(4), pp. 255-268.

Cohen, S., Mamou, J., Kanza, Y., & Sagiv, Y. (2003). XSEarch: A Semantic Search Engine for XML. *In Proceedings of the 29th International Conference on Very Large Data Bases (VLDB 2003)*, pp. 45-56. Berlin, Germany.

Contreras, J., Benjamins, V. R., Blázquez, M., Losada, S., Salla, R., Sevilla, J., et al. (2004). A Semantic Portal for the International Affairs Sector. *In Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004),* pp. 203-215. Whittlebury Hall, UK.

Croft. (2000). Combining approaches to information retrieval. *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, pp. 1-36. Kluwer Academic Publishers.

Croft. (1986). User-specified domain knowledge for document retrieval. *In Proceedings of the 9th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1986)*, pp. 201-206. Pisa, Italy.

d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., & Motta, E. (2007). Watson: A Gateway for Next Generation Semantic Web Applications. *In Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*. Busan, South Korea.

d'Aquin, M., Gridinoc, L., Sabou, M., Angeletou, S., & Motta, E. (2007). Characterizing Knowledge on the Semantic Web with Watson. *In Proceedings of the 5th International EON Workshop at International Semantic Web Conference (ISWC 2007)*. Busan, South Korea.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6), pp. 391-407.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., et al. (2003). A Case for Automated Large Scale Semantic Annotation. *Journal of Web Semantics*, 1(1), pp. 115-132.

Ding, L., Finin, T., Joshi, A., Pan, R., & Cost, S. (2004). Swoogle: A Search and Metadata Engine for the Semantic Web. *In Proceedings of the 13th Conference on Information and Knowledge Management (CIKM 2004)*, pp. 625-659. Washington DC, NY, USA.

Dumais, S. (1990). Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. TM-ARH-017527. Bellcore.

Fernández, M., Cantador, I., & Castells, P. (2006). CORE: A Tool for Collaborative Ontology Reuse and Evaluation. *In Proceedings of the 4th International Workshop on Evaluation of Ontologies for the Web (EON 2006), at the 15th International World Wide Web Conference (WWW 2006).* Edinburgh, UK.

Fernández, M., Vallet, D., & Castells, P. Probabilistic Score Normalization for Rank Aggregation. *In Proceedings of the 28th European Conference on Information Retrieval (ECIR 2006)*, pp. 553-556. London, UK.

Fernández, M., López V., Sabou, M., Uren V., Vallet, D., Motta, E., & Castells, P. (2008) Semantic Search meets the Web. *In Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC 2008)*, pp. 253-260. Santa Clara, CA, USA.

Fernández, M., López, V., Motta, E., Sabou, M., Uren, V., Vallet, D., & Castells, P. (2009) Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale. *In Proceedings of the Semantic Search Workshop at 18th International World Wide Web Conference (WWW 2009)*, Madrid, Spain.

Finin, T., Mayfield, J., Fink, C., Joshi, A., & Cost, R. S. (2005). Information retrieval and the semantic Web. *In Proceedings of the 38th Annual Hawaii international Conference on System Sciences (HICSS 2005)*, pp. 4-4. Big Island, HI, USA.

Giunchiglia, F., Kharkevich, U., & Zaihrayeu, I. (2009). Concept Search. *In Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pp. 429-444. Heraklion, Greece.

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarrán, J. (1998). Indexing with WordNet synsets can improve Text Retrieval. *In Proceedings of the COLING/ACL Workshop on Usage of WordNet for Natural Language Processing.* Montreal, Canada.

Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E., & Mena, E. (2007). Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. *In Proceedings of the 2007 Ontology Matching Workshop, at 6th International and 2nd Asian Semantic Web Conference (ISWC/ASWC 2007).* Busan, South Korea.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 199-220.

Guha, R. V., McCool, R., & Miller, E. (2003). Semantic search. *In Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, pp. 700-709. Budapest, Hungary.

Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-cream – Semi-automatic Creation of Metadata. *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management – Ontologies and the Semantic Web (EKAW 2002)*. pp. 358-372. Siguenza, Spain.

Harbourt, A. M., Syed, E. J., Hole, W. T., & Kingsland, L. C. (1993). The ranking algorithm of the Coach browser for the UMLS Metathesaurus. *In Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care*, pp. 720-724. Washington D.C., NY, USA.

Hawking, D. (2000). Overview of the TREC-9 Web Track. In *SIRO Mathematical and Information Sciences*, pp. 87-87.

Hersh, W. R., & Greenes, R. A. (1990). SAPHIRE – An information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships. *Computers and Biomedical Research*, 23, pp. 410-425.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic Annotation, Indexing, and Retrieval. *Journal of Web Semantics*, 2(1), pp. 49-79.

Lee, J. H. (1997). Analysis of multiple evidence combination. *In Proceedings of the 20th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1997)*, pp. 267-276. New York, NY, USA.

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

Lewis, D. D., & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. *In Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*, pp. 3-12. Dublin, Ireland.

Lloyd, S. M, & Roget, P. M. (1982). Roget's thesaurus of English words and phrases. Longman, Harlow, Essex.

Lopez, V., Sabou, M., & Motta, E. (2006). PowerMap: Mapping the Real Semantic Web on the Fly. *In Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pp. 414-427. Athens, GA, USA.

López, V., Sabou, M., Uren, V., & Motta, E. (2009) Cross-Ontology Question Answering on the Semantic Web – an initial evaluation. *In Proceedings of the Knowledge Capture Conference, (K-CAP 2009)*, pp. 17-24.California, CA, USA.

Luke, S., Spector, L., & Rager, D. (1996). Ontology-Based Knowledge Discovery on the World-Wide Web. *Internet-Based Information Systems: Papers from the AAAI Workshop. AAAI*, pp. 96-102. Menlo Park, CA, USA.

Madala, R., Takenobu, T., & Hozumi, T. (1998). The use of WordNet in information Retrieval. *Use of WordNet in Natural Language Processing Systems*, pp. 31-37. Montreal, Canada.

Maedche, A., Staab, S., Stojanovic, N., Studer, R., & Sure, Y. (2003). SEmantic portAL: The SEAL Approach. *Spinning the Semantic Web,* pp. 317-359.

McCool, R., Cowell, A. J., & Thurman, D. A. (2005). End-User Evaluations of Semantic Web Technologies. *In Proceedings of the ISWC 2005 Workshop on End User Semantic Web Interaction.* Galway, Ireland.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the Web. *Technical Report.* Stanford InfoLab.

Ponzetto, S. P., & Poesio, M. 2009. State-of-the-art NLP approaches to coreference resolution: theory and practical recipes. *In Tutorial Abstracts of ACL-IJCNLP 2009*, pp. 6-6. Morristown, NJ, USA.

Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., & Kirilov, A. (2004). KIM – A Semantic Platform for Information Extraction and Retrieval. *Journal of Natural Language Engineering*, 10(3-4), pp. 375-392.

Prud'hommeaux, E., & Seaborne, A. (2006). *SPARQL Query Language for RDF.* W3C Working Draft.

Sabou, M., Gracia, J, Angeletou, S., d'Aquin, M., & Motta, E., (2007). Evaluating the Semantic Web: A Task-based Approach. *In*

Proceedings of the 6th International Semantic Web Conference (ISWC 2007), pp. 423-437. Busan, South Korea.

Salton, G. (1986). Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill.

Seaborne, A. (2004). *RDQL – A Query Language for RDF*. W3C Member Submission.

Shaw, J. A., & Fox, E. A. (1993). Combination of multiple searches. *In Proceedings of the 1993 Text REtrieval Conference*, pp. 243-252.

Shoval, P. (1981). Expert/consultation system for a retrieval database with semantic network of concepts. *In Proceedings of the 4th Annual International ACM SIGIR Conference on Information storage and retrieval: theoretical issues in information retrieval (SIGIR 1981)*, pp. 145-149. Oakland, CA, USA.

Spärck Jones, K. (1964). Synonymy and Semantic Classification. *Ph.D. thesis*. University of Cambridge, Cambridge, UK.

Stojanovic, N. (2003). On Analysing Query Ambiguity for Query Refinement: The Librarian Agent Approach. *In Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003)*, pp. 490-505. Chicago, IL, USA.

Stojanovic, N., Studer, R., & Stojanovic, L. (2003). An Approach for the Ranking of Query Results in the Semantic Web. *In Proceedings of the 2nd International Semantic Web Conference (ISWC 2003)*, pp. 500-516. Sanibel Island, FL, USA.

Sure, Y., & Iosif, V. (2002). First Results of a Semantic Web Technologies Evaluation. *Common Industry Program at the federated event: ODBASE 2002 Ontologies, Databases and Applied Semantics*. Irvine, CA, USA.

Todorov, D., & Schandl, B. (2008). Small-Scale Evaluation of Semantic Web-based Applications. University of Vienna, Vienna, Austria.

Van Rijsbergen, C. J. (1979). Information Retrieval. Butkrworthe, London, UK.

Vorhees, E. (1994). Query expansion using lexical semantic relations. *In Proceedings of the 17th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*, pp. 61-67. Dublin, Ireland.

Zaragoza, H., Rode, H., Mika, P., Atserias, J., Ciaramita, M., and Attardi, G. (2007). Ranking very many typed entities on wikipedia. *In Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, pp. 1015-1018. New York, NY, USA.