Semantic retrieval and ranking of Semantic Web documents using free-form queries

Vassilis Spiliopoulos, Konstantinos Kotis*
and George A. Vouros

Ai-Lab, Department of Information and Communication Systems Engineering,
University of the Aegean, Samos, Greece
E-mail: vspiliop@aegean.gr
E-mail: kotis@aegean.gr
E-mail: georgev@aegean.gr
*Corresponding author

Abstract: Recent advances in the Semantic Web research community actuated the experimentation with a variety of approaches concerning the retrieval of Semantic Web Documents (SWDs). Most approaches require that queries are formed in a structured and formal way. Given the inability of the majority of web users to express formal queries, this paper proposes an approach to SWDs' retrieval, aiming to support users to place queries, requiring no knowledge and skills for expressing these queries in a formal language. The paper discusses specific issues that are considered towards the retrieval of SWDs using free-form queries and presents an evaluated two-step approach.

Keywords: semantic search; semantic web document; SWD; query-ontology; semantic retrieval; formal queries.

Reference to this paper should be made as follows: Spiliopoulos, V., Kotis, K. and Vouros, G.A. (2008) 'Semantic retrieval and ranking of Semantic Web documents using free-form queries', *Int. J. Metadata, Semantics and Ontologies*, Vol. 3, No. 2, pp.95–108.

Biographical notes: Vassilis Spiliopoulos holds a BSc in Computer Science from the National and Kapodistrian University of Athens, Greece, Department of Informatics and Telecommunications. Currently, he is pursuing a PhD within the Department of Information and Communication Systems Engineering (Artificial Intelligence Laboratory) of the University of the Aegean, Greece, in collaboration with the National Center of Scientific Research 'Demokritos', Institute of Informatics and Telecommunications (Software and Knowledge Engineering Laboratory). His PhD thesis concerns ontology alignment. His research interests include ontology engineering, machine learning, multi-agent systems and ontology-based information extraction.

Konstantinos Kotis holds a BSc in Computation from the University of Manchester, UK (1995), and a PhD in Information Management from the Department of Information and Communication Systems Engineering, University of the Aegean, Greece (2005). Currently, he is a researcher of the AI Lab of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece, and Director of the Information Technology Department of Local Authorities – Prefecture Administration in Samos, Greece. His research and published work concerns ontology engineering and Semantic Web (grid, search, OE, services) technologies. His published scientific work includes more than 20 book chapters, journal and national and international conference papers. He has served as a member of programme committees of international conferences on related topics.

George A. Vouros holds a BSc in Mathematics and a PhD in Artificial Intelligence both from the University of Athens, Greece. Currently, he is Professor and Dean of the School of Sciences, University of the Aegean, Greece, Director of the AI Lab and president of the Hellenic Artificial Intelligence Society. He has done research in the areas of expert systems, knowledge management, collaborative agents, ontologies, and agent-based systems. His published scientific work includes more than 80 book chapters, and journal and national and international conference papers. He has served as programme chair and chair and member of organising committees of national and international conferences on related topics.

1 Introduction

Current keyword-based web search engines provide access to billions of indexed webpages for thousands of people. Phenomena such as polysemy (one word with several meanings) and synonymy (several words with one meaning) of words increase the amount of irrelevant results that may be provided as answers to a query. Therefore, there is a great need for the meticulous treatment of the way webpages are authored and retrieved, even if topic hierarchies (e.g., in Yahoo[1]), or controlled languages are being used. Semantic Web technologies provide a potential for solving this problem.

The Semantic Web is an extension to the traditional web, aiming to facilitate information searching and sharing by exploiting ontologies and other formal artefacts that are being built using standard languages (e.g., OWL[2]). Since conventional web search engines cannot exploit documents' contents adequately, e.g., by retrieving documents related to the meaning rather than to the lexicalisation of concepts, the Semantic Web research community proposes the use of semantic search engines (e.g., OntoSearch[3]) and several other semantic search technologies (e.g., Semantic Portals (Zhang et al., 2005), Semantic Wikis (Völkel et al., 2006), multi-agent P2P ontology-based semantic routing (of queries) systems (Tamma et al., 2004), and ontology-matching-based query/answering systems (Straccia and Troncy, 2006; Lopez et al., 2006a; Kotis and Vouros, 2006; Bouquet et al., 2004)). Using these technologies, queries must be described in a formal way, so that a semantic matching algorithm to retrieve those web documents that match the query.

Although the Semantic Web technology contributes much to the retrieval of web information, there are some open issues to be tackled. First of all, the huge number of unstructured web documents must be semantically annotated to be used by semantic search technologies. This is not an easy task, as it requires, among others, the development of domain-specific ontologies. A fully automatic annotation process is still an open issue. On the other hand, the effective retrieval of web documents requires, beyond the existence of ontologies, the construction of formal queries. This is difficult, given that ordinary web users must learn a formal language for the construction of formal queries. Techniques towards automating the transformation of a free-form query (e.g., formed as a natural language sentence, or as a set/list of keywords) to a formal one are currently under study (Kotis and Vouros, 2006; Lopez et al., 2006a). The mapping of domain ontologies to formal queries constructed in the form of an ontology are also under study (Straccia and Troncy, 2006; Lopez et al., 2006a; Kotis and Vouros, 2006; Bouquet et al., 2004). The aim is towards retrieving heterogeneous and distributed web documents, since in open environments like the World Wide Web, the schema (i.e., the semantic representation of the content) of these documents cannot be known a priori (Bouquet et al., 2004; Straccia and Troncy, 2006).

A Semantic Web Document (SWD) can be considered as a document whose content is either an ontology (also known as schema or model) or a simple conventional web document annotated with specific tags taken from a domain ontology (Ding et al., 2004). In fact, the annotated conventional web documents can be divided into a variety of different types according to the type of the ontology used to annotate the document, i.e., lightweight or heavyweight ontology (Corcho, 2006). In this paper, we assume that SWDs are ontologies, not annotated documents. These documents are also referred as 'Semantic Web ontologies' (Ding et al., 2004) and may also include instance data. In this paper, we are concerned with the problem of retrieving this kind of documents found in an SW repository such as Swoogle, by transforming free-form queries into formal ones (called 'query-ontologies'). The aim is to facilitate the semantic retrieval by matching query-ontologies to OWL documents (ontologies) of a Semantic Web repository and ranking the retrieved documents according to their relevance to the query-ontology. We call this type of retrieval ontology-matching-based retrieval of SWDs. Having said that, we conjecture that conventional web documents annotated with heavyweight ontologies such as OWL ontologies (Corcho, 2006), can be also retrieved by our approach given that their annotation tags are valid URIs of the referenced ontology. Our approach cannot be used to retrieve instance data from conventional web documents that have been annotated with lightweight ontologies.

The problem of semantically searching heterogeneous and distributed SWDs has been well defined and methodologically addressed by other recent step-by-step approaches (Straccia and Troncy, 2006; Lopez et al., 2006a). Thus, this paper will not consider methodological issues towards solving this problem. The aim of the paper is to present an evaluated implementation of a two-step approach to SWDs' retrieval, namely Semantics and Automated Matching of Ontologies in Search (SAMOS).

However, going beyond the work of other approaches, the proposed approach focuses on supporting users to place free-form queries, requiring no knowledge and skills for expressing these queries in a formal language (Lopez et al., 2006a). The paper also discusses additional important issues towards approaching the retrieval of SWDs in such a manner. Furthermore, the paper presents an enhanced method of our proposed approach focusing on the retrieval of SWDs using free-form queries, without requiring any external resources. Preliminary but promising results are presented in this direction.

This paper is structured as follows: Section 2 provides background knowledge and related work concerning web search and semantic search of heterogeneous and distributed SWDs. Section 3 discusses additional issues to methodological ones, towards approaching the semantic search of SWDs using free-form queries. Section 4 presents the proposed system for querying and retrieving/ranking SWDs utilising the Swoogle semantic search engine, and Section 5 presents an enhanced method of the proposed approach. Finally, we conclude with a summary of remarks and future implementation plans.

2 Background and related work

A keyword-based web search method integrates techniques that are based on string (lexical) matching of the query terms with the terms contained in web documents. Traditionally, keyword-based search is being used for the retrieval of web documents, without making the meaning of these keywords explicit in any formal way. There are several techniques that have been proposed for keyword-based search over the web (Alesso, 2004): simple Boolean search (combining keywords using the Boolean operators AND, OR and NOT), wildcard and proximity search, fuzzy search, contextual search, keyword location-based search, human (or topic)-directed search, thesaurus-based search, statistics-based search such as Google's PageRank[4] technology.

Keyword-based search technology has also been used to retrieve SWDs by matching query terms to terms that lexicalise ontology elements in an SWD. Such a technique is being realised by the Swoogle search engine (keyword-based retrieval of SWDs). However, this technology does not exploit the semantics of the SWDs, and therefore, does not make the best of the available information. In the most general case, semantic search must be formed as an extension of the keyword-based one, where the syntactic similarity between terms, although it may provide an evidence for semantic matching, it is not of direct interest. In fact, what is important is the similarity of terms' meaning. For instance, a match between the query-term 'book' and a document-term 'reserve' may be correctly identified if the meaning of the term 'book' is "the reservation of a ticket". On the other hand, a match of the term 'book' with an identical term found in a web document may be incorrectly identified, if their meanings are completely different: The query-term 'book' may denote any publication, and the document-term 'book' may denote any reservation.

The semantic retrieval of the SWDs requires the semantic matching of query terms and document terms. If the query is formally specified, the semantics of each term are explicitly defined. For instance, if a query is specified by means of an ontology, then the semantics of each term lexicalising an ontology concept is revealed by exploiting the semantic relations between this concept and other ontology elements. On the other hand, if the query is informally specified, e.g., it is formed in natural language or as a list of keywords, the semantics of each term in the query must be somehow uncovered. The issue here is how a machine can 'guess' the intended meaning of an informal query to retrieve the document that is closer to it, and therefore more interesting to the user.

Furthermore, even if a query is formally specified, to be able to compute a semantic matching, the content of documents must also be explicitly and formally specified. In case of an SWD, the semantics of the document are formally and explicitly specified by means of an ontology. In case of unstructured documents, advanced ontology learning techniques can be used to annotate them and further extract their meaning: This is an issue that this paper does not deal with, as it supposes that documents have somehow been tagged in a formal way using heavyweight ontologies (Corcho, 2006).

Concerning the meaning of queries, there are several proposals in the literature. Web-focused approaches and intelligent search engines such as AskJeeves[5] (Teoma technology) try to 'guess' the intended meaning of an informal query by analysing the terms and their relations in a sophisticated way using natural language processing techniques, or by refining the query in collaboration with the users. An alternative technique (Kotis and Vouros, 2006) maps each term of a query to its intended meaning (sense found in a lexicon such as WordNet (Miller, 1995)) using Latent Semantic Indexing (LSI) techniques (Deerwester et al., 1990). Close to our aims is also the work presented in Karanastasi and Christodoulakis (2007), where an ontology-driven semantic ranking methodology is being used for the disambiguation of a natural-language query. This work has been proposed in the context of the OntoNL framework. The disambiguation procedure is automatic and quite promising.

Concerning the semantic retrieval of SWDs, although there are several approaches, the majority of them assume that the query is already available in a structured way using a formal language, e.g. Bouquet et al. (2004), Straccia and Troncy (2006). On the other hand, the use of a controlled language, instead of a formal one, to formulate a semantic query (Bernstein et al., 2005) provides a solution to a certain respect, but it still requires that the users should learn a new language. The retrieval of SWDs using formal queries may by done either in cases where the SWDs concern the same ontology (i.e., semantic homogeneity) or in cases where the SWDs concern different ontologies (semantic heterogeneity). In the former cases, one may use a Semantic Web query language such as OWL-QL (Fikes et al., 2003) or RQL (Karvounarakis, 2003), while in the latter cases the retrieval can be performed by either using a shared common ontology where queries and SWDs are being mapped, or by performing horizontal mappings across local SWDs using ontology-matching techniques. Two recent ongoing approaches that propose solutions for retrieving heterogeneous and distributed SWDs, PowerAqua (Lopez et al., 2006a) and oMap (Straccia and Troncy, 2006), use ontology-matching and query reformulation techniques. Both approaches aim at implementing systems for ontology-matching-based SWDs retrieval, following a similarapproach to the one presented in this paper.

oMap provides a well-defined step-by-step methodological approach towards solving the problem of SWDs' retrieval by clearly identifying the steps of query reformulation and ontology matching. However, it does not provide a solution for dealing with free-form queries. We conjecture that a Semantic Web Search System (SWSS) should not consider the use of a formal language or any other special-purpose controlled language as a necessity for the formulation of queries. Ordinary users are not willing to learn a new language, especially if this is far from their speaking one. In addition, users do not want to change the way they search information in their everyday tasks. Thus, an SWSS should be able to accept queries in free form, i.e., either in natural language or as a set/list of keywords/terms (there are no formal rules for the construction of the query).

On the other hand, PowerAqua (which extends AquaLog (Lopez et al., 2005) by providing answers drawn from multiple, heterogeneous and distributed ontologies over the web), following similar to oMap methodological steps towards approaching the problem (query reformulation and ontology matching), provides means for placing queries in natural language. The input query of this approach is disambiguated using external resources and transformed into a triples-formed query-ontology. An important limitation is that the system, to be able to perform the process, must use a reference ontology for every domain, and also the WordNet lexicon for term disambiguation. The use of a reference ontology is usually a hindrance to the whole process since it is very often the case that such knowledge is just not available (at the time users need it) in open and continuously evolving environments such as the Semantic Web, or if it is available, it is not easy to extract and use the semantics needed for the reformulation of the query. We must point that in most of the cases, domain-specific terminology does not need disambiguation, since concepts have very domain-specific lexicalisations with very narrow meaning. However, when needed, a generic lexicon such as WordNet seems to be adequate. Furthermore, for the matching of the query to the SWDs, PowerAqua uses an algorithm namely PowerMap (Lopez et al., 2006b), which is still under implementation.

Both related approaches presented in this section, oMap and PowerAqua, provide similar methodological steps towards solving the problem of Semantic Web search, and we do follow similar ones:

- query reformulation
- ontology matching.

This paper presents the implementation of a SWSS based on these methodological steps. In contrast to oMap, the proposed approach allows for a query to be placed in a free form, and in contrast to the PowerAqua, the proposed approach does not necessarily require a reference ontology or lexicon during the process, aiming to the automatic and transparent retrieval and ranking of SWDs. To evaluate the efficiency and effectiveness of the presented approach, we have implemented a system (namely, SAMOS) that utilises the Swoogle search engine by performing a re-ranking of the keyword-based retrieved SWDs. The implemented system, using a semantic approach, successfully filters and re-ranks SWDs by applying an automated ontology matching tool (AUTOMS (Kotis et al., 2006b)), and free-form query terms' disambiguation techniques. The SWDs that are semantically most relevant to the free-form query are placed at the top of the result list. Finally, the paper discusses preliminary results of an enhanced method of the proposed approach, where no external resources are necessary.

3 Issues concerning the retrieval of SWDs

Latest research efforts have proposed and demonstrated that the retrieval of SWDs in distributed and heterogeneous settings must be performed by utilising ontology matching systems (Straccia and Troncy, 2006; Lopez et al., 2006a; Kotis and Vouros, 2006). In this section, taking into account methodological issues proposed in the PowerAqua and oMap approaches, we discuss additional issues that we consider to be important when designing and implementing a semantic search system towards retrieving SWDs.

- *Use of external resources*. A SWSS, as PowerAqua approach proposes, needs to incorporate additional/external resources if it uses free-form queries. Such knowledge can be in the form of a generic lexicon/thesaurus or/and in the form of a reference ontology. In this paper, we conjecture that any external resource that provides information concerning the semantics of terms can be used to disambiguate a free-form query and to produce the triples needed for the semantic matching of the reformulated query to the SWDs. Having said that, we also conjecture that in most of the cases, domain-specific terminology does not need disambiguation since terms have a very narrow meaning. Based on this assumption, in our ongoing work, we are experimenting with mapping algorithms that directly map free-form queries to SWDs, without presupposing the existence of any external resource.
- *Automation and transparency*. An SWSS should provide the whole process transparently to the end-user, delivering a ranked list of SWDs that match each query. Querying SWDs should be performed with the minimum human involvement. Users should be only involved in validating the output of the semantic search method. In this paper, we present methods and tools that are fully automated and transparent to the end-user.
- *Performance*. Retrieval of SWDs must be performed promptly. The response time in real-time querying environments such as in a Semantic Web search engine is of high importance. Thus, an SWSS should be implemented as a few-steps-process, acquiring short time for executing each step for delivering the desired result. Currently, we are not considering the overall performance of the implemented system, but rather the performance of individual components. Having said that, in Section 5 we present an enhanced method of the proposed approach, which, among other advantages, reduces the execution time of semantic retrieval significantly since

- the query reformulation step is not necessary
- only a single mapping method is applied between the query and the SWDs, instead of the combination of mapping methods applied with AUTOMS ontology mapping tool
- only a fragment (matched concepts and their vicinity) of the SWDs is considered in the mapping method instead of the large SWDs found in repositories.

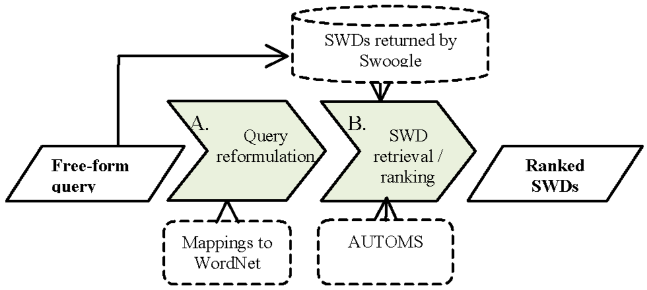
- *Precision/Recall*. The accuracy of an SWSS is also an important issue. Approaching the querying of SWDs in real settings, the technologies and implementations used for developing an SWSS should be tested and evaluated with respect to precision and recall. Specifically, the automatic disambiguation of terms (automatic assignment of lexicon senses to terms) and the automatic retrieval of SWDs (matching a query-ontology to SWDs) should provide high precision and recall measurements.

4 Implementation of the SAMOS SWSS

In this section, we present a specific implementation of an SWSS, namely SAMOS, developed for experimental purposes. SAMOS actually combines several technologies towards delivering a meta-engine for filtering SWDs returned by the Swoogle search engine. Swoogle (Ding et al., 2004) is a crawler-based indexing and retrieval system for SWDs in RDF(S), DAML, or OWL syntax. Swoogle provides techniques for semantically relating SWDs prior to the execution of queries. It extracts metadata and computes relations between documents. Although Swoogle in its present status serves as an SWD indexing system, the retrieval technology used is based on the lexical matching of query terms and the indexed labels of ontology classes and properties. By using Swoogle, the aim is to prove that the precision of the retrieval for a simple query can be improved if the proposed semantic search method is applied.

SAMOS (see Figure 1) implements the automatic construction of formal queries (query-ontologies) from free-form queries towards the ontology-matching-based retrieval/ranking of SWDs. As already mentioned, other proposed systems (or approaches) use either formal languages to construct the query, or use an existing domain ontology as a pre-built query, or use a pre-existing reference ontology in addition to a free-form query for on-the-fly meaning disambiguation. Our basic implementation falls in the third case, where external resources (lexicon) are used for disambiguating the meaning of the free-form query. However, this implementation does not require the existence of a reference ontology: The external resources are embedded in the disambiguation method by means of the generic lexicon WordNet. As already mentioned, this is an advantage over other solutions since it is not always the case (in fact it is very rare) that a reference ontology will exist for every domain that need to be queried. The proposed basic method uses the WordNet generic lexicon to disambiguate query terms by automatically mapping them into WordNet senses. As already stated, any other lexicon or thesaurus that can provide semantic relations between the query terms such as subsumption, equivalence, part-of, etc., can be used for this implementation. An enhanced method of SAMOS approach, presented in Section 5, does not require the use of any external resource.

Figure 1 The SAMOS system overall architecture based on a two-step approach (steps are depicted as block-arrows) (see online version for colours)



In addition to the automatic construction of the query-ontology, SAMOS implements the retrieval of SWDs using the automated ontology-matching tool AUTOMS (Kotis et al., 2006b). For the effective retrieval of SWDs, AUTOMS computes the similarity between the SWDs and the reformulated query (i.e., the query-ontology). AUTOMS extends the HCONE-merge method (Kotis et al., 2006a) by combining lexical, semantic, and structural matching methods. Lexical matching computes the matching of ontology concept names (labels at nodes), estimating the similarity among concepts using syntactic similarity measures. Structural matching computes the matching of ontology concepts by taking into account the similarity of concepts in their neighbourhoods. The neighbourhood of a concept includes those concepts that are related to it. Finally, semantic matching concerns the matching between the meanings of concept specifications. The computation of semantic matching may rely to external information found in lexicons, thesauri or reference ontologies, incorporating semantic knowledge (mostly domain-dependent) into the process.

Finally, SAMOS system implements the ranking of retrieved SWDs based on how well they match to the query-ontology: This is determined by the number of mappings computed between the query-ontology and an SWD. The more are the mappings between the query-ontology and an SWD, the highest is the position of that SWD in the final ranking. Actually, the set of SWDs that participate in the ranking algorithm is the set of documents retrieved by the submission of the free-form query to Swoogle. Thus, the ranking can be seen as a filtering process of SWDs returned by a keyword-based search.

In what follows, a more detailed step-by-step description of SAMOS implementation is presented, outlining technological issues for each individual step of the approach, considering also example queries.

4.1 Step A: Query reformulation

In this step, each term of the free-form query is disambiguated, assessing its user-intended meaning, which is specified by a WordNet sense. Although in other lines of our research this process is accomplished by the utilisation of LSI technology (Kotis et al., 2006a), in this implementation we have used Vector Space Model (VSM) technology (Raghavan and Wong, 1986) due to the nature of available data (i.e., very few terms in queries) and the need to reduce the response time of SAMOS. In what follows, we present the specific VSM implementation in detail and we also show how its output is being used to construct triples in the form of a query-ontology, reformulating the initial free-form query into a formal one.

4.1.1 Query disambiguation

To map a query term to its intended meaning, we compute the semantic similarity of this term to a set of WordNet senses. The set of WordNet senses is obtained from the lexical matching of the term with a WordNet term entry. The algorithm takes into account the vicinity V_t of each query term t. Since this computation is based on the hypothesis that query terms are related to each other, the vicinity of a query term includes all the other terms in the query. In what follows, we describe in detail how VSM is exploited to disambiguate a free-form query using the WordNet lexicon.

Vector Space Model (VSM)

A query term t is represented as a document ('bag of words' representation). Since a term t is related with all terms in its vicinity (V_t), the document representing t includes all terms occurring in V_t. Similarly, each WordNet sense $S_1, S_2, \dots, S_m$, representing the m possible meanings of the term t, is represented as a document.

In our case, we adopt the most common document representation in the field of Information Retrieval, which is a weighted vector of the form $(w_1, w_2, \dots, w_N)$, where w_i, $i = 1, \dots, T$ is the *tf-idf* value (Raghavan and Wong, 1986) of the corresponding word i (of the distinct T terms extracted from all the WordNet senses plus all query terms in the vicinity V_t of t), extracted from the related WordNet senses or a query term in the vicinity V_t of t. The w_i of a term i is calculated as follows:

$$w_i = tf_i \times idf_i$$

$$idf_i = \log_2 \frac{N}{n_i}$$

where tf_i (*t*erm *f*requency) is the number of times that the term i appears in a particular document (query or WordNet sense), idf_i (*i*nverse *d*ocument *f*requency) is the inverse of the percentage of the documents that contain the word i, N is the total number of documents and n_i is the number of documents that contain the word i at least one time. The major advantage of the utilisation the *tf-idf* technique is that it identifies and promotes terms that are discriminative for documents in the corpus. The word weight gives prominence to the words closely related to the specific documents.

It must be pointed out that in the case of WordNet utilisation, the intended meaning of a term t is computed using VSM against all available senses ($S_1, S_2, \dots, S_m$) of the corresponding WordNet entry. Terms are extracted from a WordNet sense S using the following information:

- words that label the sense
- natural language description of the sense, namely the 'gloss'
- all direct hypernyms and hyponyms of S.

Tokenisation, stemming and elimination of stop words are performed on the set of extracted terms.

As already mentioned, there are cases were a reference ontology may be used instead of WordNet lexicon as an external resource for disambiguation of very specialised domain queries (e.g., medical). In that case, the terms that will be used in the VSM documents are extracted from the reference ontology using the following information:

- names, labels and comments of an ontology concept
- names, labels and comments of the concept properties
- names, labels and comments of the concept instances
- names, labels and comments of all directly related concepts via subsumption or other types of relations.

Tokenisation, stemming and elimination of stop words are performed on the set of extracted terms.

The mapping of a query term to a document (sense in case of WordNet or set of concept names/labels/comments in case of a reference ontology) is computed by measuring the distance between the query vector q and each document vector. The result is a ranked list of documents. The document with the highest cosine coefficient similarity (Salton and McGill, 1983) represents the user-intended meaning of term t. The cosine coefficient similarity between two vectors $\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{iT})$ and $\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jT})$ is defined as follows:

$$\mathrm{Sim}(w_i, w_j) = \frac{\sum_{k=1}^{T} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{T} (w_{ik})^2 \times \sum_{k=1}^{T} (w_{jk})^2}}$$

The steps for mapping a query term to a WordNet sense using *VSM* are shown in Figure 2:

Figure 2 The steps for computing the mapping of a term to a WordNet sense

1. Choose a term from the query string. Let *t* be the term name.
2. Get all WordNet senses $S_1, S_2, \ldots, S_m$, lexicalised by *t*.
3. Get the direct hyperonyms and hyponyms of all *t* senses.
4. For each WordNet sense $S_1, S_2, \ldots, S_m$ create a corresponding document, based on the *VSM*.
5. Build a document for the query term *t* using all terms in the query string (i.e. *t* and its *vicinity* V_t), based on the *VSM*.
6. Find the ranked associations between the query document of term *t* and the documents representing the *WordNet* senses of *t*, and consider the association with the highest cosine coefficient similarity.

Discussion and examples

After experimentation with other VSMs (e.g., LSI), we found that the VSM delivers better results when the query is constructed by only a few terms. We have tested several queries with different number of terms. We have concluded by experiments that queries using three and more terms deliver better results.

Consider the following example query, "*play theater mystery*". A human could guess that the user-indented meaning of the term 'play' is captured by the WordNet sense

> "play, drama, dramatic play – (a dramatic work intended for performance by actors on a stage; "he wrote several plays but only one was produced on Broadway")."

The VSM-based disambiguation method can also guess the intended meaning of the term 'play' by automatically generating the mapping of this term to the corresponding sense. Furthermore, the rest of the terms are also correctly mapped to the appropriate WordNet senses: 'theater' is mapped to *"dramaturgy, dramatic art, dramatics, theater, theatre – (the art of writing and producing plays)" and 'mystery' to "mystery, mystery story, whodunit – (a story about a crime (usually murder) presented as a novel or play or movie)".*The WordNet version (2.0) that we exploited provides 57 different senses for 'play', two for 'mystery' and three for 'theater'.

For experimentation purposes, we have been tuning/refining queries by including other terms also. In the case of "play a role in theater", for instance, the term 'play' has a meaning, which is different from that mentioned above: "to play a role in a theatrical play". The newly introduced term in the freely formed string, i.e., 'role', has four different senses in WordNet. The system correctly uncovers the intended meaning of this term and maps the term 'play' to the WordNet sense

> "act, play, represent – (play a role or part; "Gielgud played Hamlet"; "She wants to act Lady Macbeth, but she is too young for the role"; "She played the servant to her husband's master")."

The term 'theater' has the same meaning as in the previous example, while the term 'role' is correctly mapped to the sense

> "character, role, theatrical role, part, persona – (an actor's portrayal of someone in a play; "she played the part of Desdemona")."

Considering another example query, "print a book about flight" with an intended meaning of "printing a book containing information about flights", the machine correctly maps the term 'book' to the sense

> "book – (a written work or composition that has been published (printed on pages bound together); "I am reading a good book on economics")"

and the terms 'print' and 'flight' to the senses "*print, impress – (reproduce by printing)*" and "*flight – (a scheduled trip by plane between designated airports; "I took the noon flight to Chicago")*", respectively. Then, we have examined the queries "book a flight trip" and "book a flight and reserve", having in mind a different meaning for the term 'book': the meaning of "reserving a flight in advance". This meaning is captured by the WordNet sense "*reserve, hold, book – (arrange for and reserve (something for someone else) in advance; "reserve me a seat on a flight"; "The agent booked tickets to the show for the whole family"; "please hold a table at Maxim's")*". The system correctly maps 'book' to the proper WordNet sense. The term 'flight' is mapped similarly to the previous example, while the term 'trip' is correctly mapped to the sense "*trip – (a journey for some purpose (usually including the return); "he took a trip to the shopping center")*".

4.1.2 Construction of the query-ontology

Having mapped terms to WordNet senses (their intended meanings), we can generate triples that comprise concepts and relations between them. Depending on whether a reference ontology or a generic lexicon such as WordNet is being consulted for discovering the intended meaning of query terms, different construction rules are being used for the construction of the query ontology. In more detail, for each query term mapped to a WordNet sense:

- A concept lexicalised by the word that labels the corresponding WordNet sense is created. For instance, the assessed sense of the term 'theater' is labelled as 'dramaturgy'. As a result, a concept labelled 'dramaturgy' is created (Figure 3).
- If more than one word labels the lexicon entry, then for each of them a new concept, lexicalised by the corresponding word, is created. All generated concepts are marked as equivalent, as all the terms labelling the corresponding WordNet senses are synonyms.
  The mapped sense of the term 'theater' contains four synonyms, namely: 'dramaturgy', 'dramatic art',

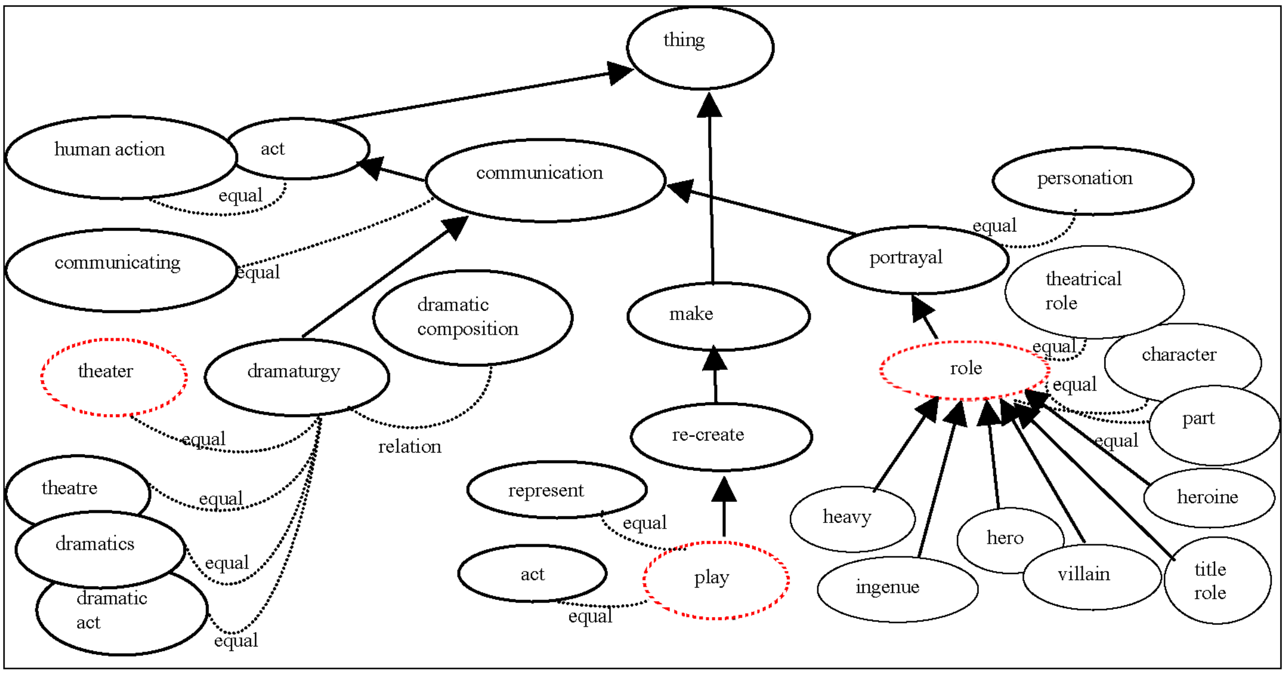
‘dramatics’, ‘theater’ and ‘theatre’. As depicted in Figure 3, four equivalent concepts are introduced.

- For all WordNet hyperonyms (hyponyms) of a WordNet sense, super-concepts (respectively, sub-concepts) of the corresponding concept that represents this particular sense are created, according to the previous rules. The resulting taxonomy for the query “*play a role in theatre*” is depicted in Figure 3. In the current implementation, two levels of hypernyms and hyponyms are exploited (not just the direct ones). In doing so, the constructed ontology includes more concepts and as a result, it leads to better performance of the implemented system.
- If two different query terms are mapped to the same WordNet sense, then this sense constitutes their common intended meaning and is represented in the query-ontology by a single concept. Moreover, if a hypernym (or hyponym) of two different query terms happens to be the same, then the concept created for this sense corresponds to the same concept in the created taxonomy. For example, as depicted in Figure 3, ‘communication’ is the hypernym of both ‘dramaturgy’ (representing the term ‘theater’) and ‘character’ (representing the term ‘role’) in the WordNet taxonomy. As a result, this is a super-concept of both the corresponding concepts in the generated ontology.
- Other kinds of semantic relations between WordNet senses (e.g., meronyms and holonyms) are represented by means of the generic property lexicalised by ‘relation’. For example, the assessed sense of the term ‘theater’ has a single direct meronym: “*dramatic composition, dramatic work – (a play for performance on the stage or television or in a movie etc.)*”. In this case, a concept is created representing this sense following the previous rules. This sense is related with ‘theater’ through a ‘relation’, as depicted in Figure 3. In this particular example, the term ‘theater’ has no holonyms.

The output of this step is a set of concepts and their relations in the form of triples, following the RDF (Schema) specifications. For the example query “play a role in theatre”, a fragment of the output triples is shown below (using space as separator between a triple’s elements):

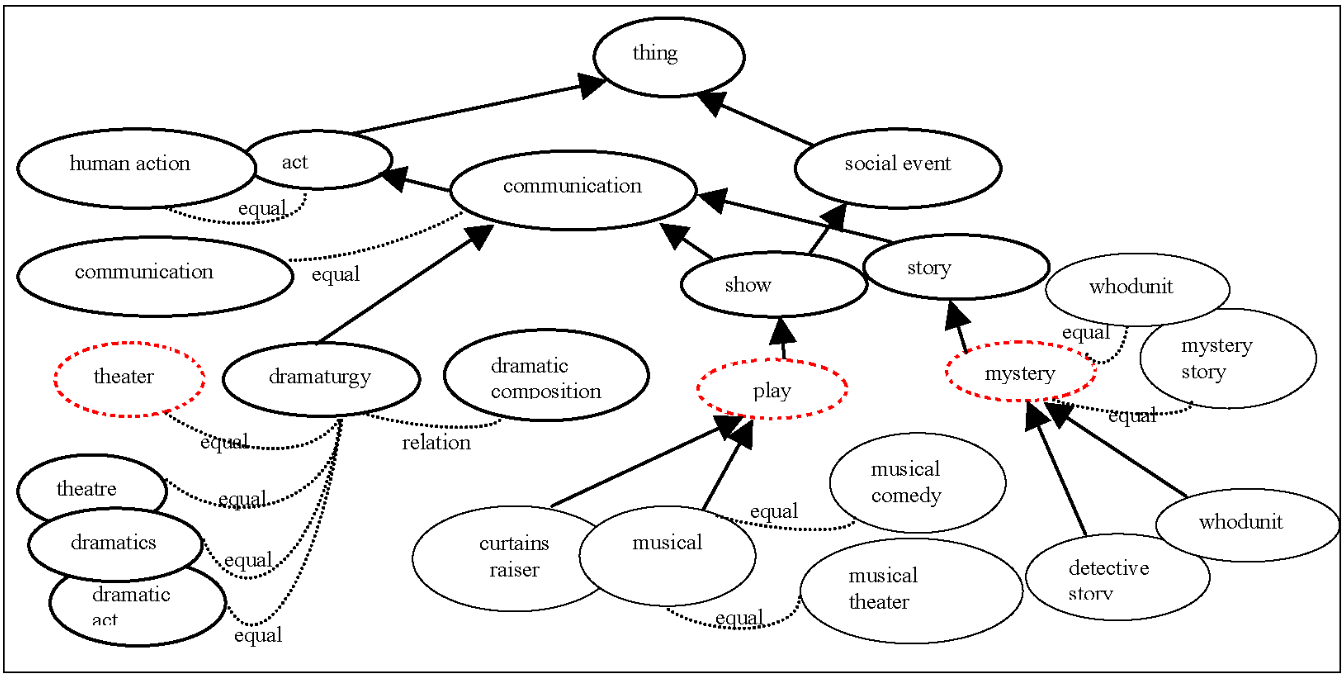
<dramaturgy rdf:type owl:Class>
<dramatic_art rdf:type owl:Class>
<dramaturgy owl:equivalentClass dramatic_art>
<dramaturgy owl:equivalentClass dramatics>
<communication rdf:type owl:Class>
<dramaturgy rdfs:subclassOf communication>
<stage rdfs:subclassOf dramaturgy>
<relation rdf:type owl:objectProperty>
<relation rdfs:domain dramaturgy>
<relation rdfs:range dramatic_composition>

Figure 3 Part of the query-ontology for “play a role in theater” query (see online version for colours)



The triples’ syntax follows the RDF (Schema) specification (Beckett, 2004) to be transformed into an OWL query-ontology in a straightforward manner. OWL builds on top of RDF and RDF Schema (Patel-Schneider et al., 2004) and adds more expressiveness for describing properties and concepts. Concerning the two example queries, “play a role in theater” and “theatre play mystery” representative fragments of the generated ontologies are depicted in Figures 3 and 4, respectively.

Figure 4 Part of the query-ontology for "theater play mystery" query (see online version for colours)



4.2 *Step B: retrieve and rank SWDs*

In this step, SAMOS re-ranks the SWDs retrieved by the keyword-based retrieval system Swoogle. The re-ranking is based on the uncovered semantics of the query terms, re-formulated as a query-ontology, and the semantics of the retrieved SWDs.

Initially, the free-form query provided by the user is placed in Swoogle to retrieve all available SWDs (ontologies) indexed by the search engine. All retrieved SWDs are mapped against the query-ontology generated in Step A. For this purpose, the AUTOMS ontology mapping tool (Kotis et al., 2006b) is utilised. The re-ranking is based on the semantic relevance of the retrieved SWDs and the query-ontology. Specifically, the relevance between two ontologies is defined as the number of their mapped concepts. The intuition behind this heuristic is the fact that concepts model the major aspects of the domain in concern. As a result, it is intuitive that the higher the number of mapped concepts, the higher is the relevance between the query-ontology and the retrieved SWDs. Retrieved SWDs with higher relevance to the query-ontology are ranked higher in the presented results' list.

AUTOMS ontology mapping tool synthesises six mapping methods: lexical, semantic, simple structural, properties-based, instances-based and the iterative structural method. AUTOMS is build using AUTOMS-F (Valarakos et al., 2007), a highly extensible and customisable Application Programming Interface (API), focusing on rapid development of automated mapping tools.

Table 1 presents the re-ranking results over Swoogle SWDs returned for the example query "play theatre role". The second column of Table 1 provides the ranking of SWDs produced by Swoogle, while the third column shows the ranking produced by SAMOS, given the query ontology shown in Figure 3. The last column shows the ratio of mappings produced to the total number of concepts in the query-ontology.

Several queries have been tested against the Swoogle engine. In Table 2, we present an example fragment of freely formed queries that have been used to evaluate the presented system. We have measured precision and recall with respect to the mappings of terms to WordNet senses (first and second column). We have also measured the re-ranking efficiency, i.e., how many documents of the returned (matched) ones were correctly re-ranked, based on the judgement provided by two separate ontology engineers (third column), given the intended meanings of queries.

The overall value of the experiment was that, based on experts' observations, the proposed system performed well concerning the re-ranking of Swoogle's SWDs. Documents that appeared low in the keyword-based results list appeared higher in the list after the re-ranking process, as an effect to their close semantic relation with the query. Query terms that have not been mapped into a correct WordNet sense resulted into the construction of partially incorrect query-ontologies and consequently into incorrect mapping to SWDs, influencing their re-ranking.

In Table 3, we further investigate our approach by performing an additional, more complex, experiment. We mainly evaluate the effectiveness of the query-ontology construction method to the overall system's performance. We start with a single-term query and progressively we augment it with more terms towards a query with specific intended meaning: "to reserve a seat in a theatrical play". Table 3 includes information concerning the size (in terms of number of defined concepts) of the created query-ontology and the number of its mapped concepts to the retrieved SWDs. This particular experiment's value can be summarised in the following observations:

- As more terms are introduced in the query, the size of the generated query-ontology increases. The number of the classes introduced in the query-ontology is dependent on the WordNet sense that each term has been mapped. Senses that have richer semantics (more hypernyms, hyponyms, etc.) as captured by WordNet lexicon introduce more concepts in the query-ontology.
- As more terms are introduced in the query-ontology, the precision of the whole query may change. The precision percentage is low or reduced when polysemous terms exist or added (see 4th, 5th and 7th row of Table 3), whereas on the other hand, the precision percentage is high or increased when terms that are related with a sense of the ambiguous terms exist or added.
- When more terms are introduced in the query-ontology, the mapping method returns related SWDs that are more focused to the query-ontology. For instance, in Table 3, query "book seat theatre play performance" returns four ontologies, with at least one mapped concept, while query "book seat theatre play performance actor" returns only one. This limits the number of related SWDs that should have been returned (recall) by a query with the same intended meaning; however, at the same time, it increases the precision of the returned SWDs.

Table 1 Precision results of query ontology "play a role in theater" over Swoogle SWDs using AUTOMS

SWDs (*ontology*)	*Swoogle ranking*	*Final ranking*	*Matched classes*
http://139.91.183.30:9090/RDF/VRP/Examples/DCD100.rdf	1	1	(21/52)
http://athena.ics.forth.gr:9090/RDF/VRP/Examples/DCD100.rdf	2	1	(21/52)
http://reliant.teknowledge.com/DAML/Mid-level-ontology.owl	3	4	(4/52)
http://reliant.teknowledge.com/DAML/Mid-level-ontology.daml	4	4	(4/52)
http://www.schemaweb.info/webservices/rest/GetRDFByID.aspx?id=241	5	5	(3/52)
http://smartweb.dfki.de/ontology/swinto0.3.1.rdfs	6	3	(6/52)
http://www.smartweb-project.org/ontology/swinto0.3.1.rdfs	7	3	(6/52)
http://www.cl.uni-heidelberg.de/kurs/ss03/ki/Uebungen/Ontologien/ontology.rdfs	8	2	(7/52)

Table 2 Examples of freely-formed queries and the re-ranking of the retrieved SWDs

Free-form query	*Num of terms mapped to a WordNet sense/num of query terms*	*Num of terms mapped correctly to a WordNet sense/num of mapped query terms to a WordNet sense*	*Number of correct SWDs re-ranked*
SWDs re-ranked			
Book a flight for domestic_trip	4/4	4/4	None (only one SWD returned)
Book a flight trip	3/3	3/3	No results obtained due to broken links or parse errors
Print a book for flight	3/3	3/3	No (only one SWD returned)
Play a role in theatre	3/3	3/3	All
Play theater mystery	3/3	3/3	All
A publication for flight	2/2	2/2	All
Book for theater play	3/3	1/3	Some (due to wrong incorrect disambiguation of 2 terms)
Book a seat for theater play	4/4	3/4	Some (due to wrong incorrect disambiguation of 1 term)
Reserve a ticket for theatre	3/3	3/3	All

Table 3 Investigation of the behaviour of a query

Free-form query	*Num of terms mapped to a WordNet sense/num of query terms*	*Num of terms mapped correctly to a WordNet sense/num of mapped query terms to a WordNet sense*	*Matched classes*
Book	1/1	0/1 (*book* is mapped to sense 'written work')	3/78, 3/78, 2/78, 2/78, 2/78, 2/78, 2/78, 1/78
Book seat	2/2	2/2	7/22, 4/22, 4/22, 1/22
Book seat theater	3/3	3/3	13/48, 4/48, 4/48, 4/48
Book seat theater play	4/4	3/4 (*play* is mapped to sense 'be performed')	13/50, 4/50, 4/50, 4/50
BOOK seat theater play performance	5/5	4/5 (*play* is again incorrectly mapped)	16/81, 6/81, 6/81, 6/81
Book seat theater play performance actor	6/6	6/6	27/251
Book seat drama theater play performance actor	7/7	6/7 (*book* is mapped to sense "script, book, playscript – a written version of a play...")	No returned SWDs (do not exist in the example repository)
Book ticket seat drama theater play performance actor	8/8	8/8	No returned SWDs (do not exist in the example repository)

The implementation of the SAMOS system has been based on the latest technological standards (OWL, JENA) as well as on state-of-the-art ontology mapping software for matching the query-ontology with the SWDs. The overall performance of this system can be evaluated in terms of time, precision and recall measures of the AUTOMS ontology matching algorithm since there are available data of other related technologies to be compared (Euzenat et al., 2006). Having said that, the response time of SAMOS system has been influenced by the size of the Swoogle SWDs, ranging (in our experiments) from few seconds to many minutes. In the case of "play a role in theater", the SWDs contain thousands of concepts (~1000–3000) and as a result the response time was 65 min. The improvement of response time is further discussed in Section 5 and is also among our concerns for future work since the scaling-up of the service is an important issue to its success.

5 Advancing the basic method

The presented SWSS, although it contributes to the overall problem of semantic search by proposing a method for the retrieval and re-ranking approach of SWDs, it does not overcome certain limitations that other systems also have. As already stated in the related work section, the need of external resources for disambiguating terms in a free-form query, also true for PowerAqua system, limits systems' accuracy since the precision and recall measures depend on the quality and extent of lexicons, reference ontologies and other resources. To overcome this limitation, we propose an alternative approach to the matching of free-form query terms against SWDs concepts, without using external resources in any stage of the overall process. In what follows, we present such an approach and comment on preliminary results.

As already described, we have been experimented query by consulting the WordNet lexicon. The output of this step, i.e., a query-ontology, is then mapped to an SWD using the AUTOMS ontology mapping tool. The alternative approach we have been lately experimenting with is as follows: We directly map free-form query terms to the terms that lexicalise SWDs' concepts lexically matching to the query terms. Although lexical matching initially identifies the relevance of the query to SWDs, by using VSM technology against the SWDs, we will end up with a new list of relevant documents utilising the semantic relations between terms that lexicalise concepts/properties of these documents. This is achieved by modifying the input of the VSM algorithm as described in Section 4. In this case, the semantic space comprises the *n* more frequently occurring terms of the *m* SWDs that Swoogle has returned. More specifically, to approach the semantic co-relation of the documents more efficiently and for speeding up the whole process, each *document* entry comprises:

- the terms that lexicalise the concepts of the SWD that have been matched to the query terms
- the terms that lexicalise all the concepts that are semantically related with the matched concepts via subsumption, equivalence, and other type of relations
- the labels and comments of the matched concepts.

The query document (vector) is actually constructed by the terms of the free-form query. So, by executing the VSM algorithm for a set of SWDs, the method will provide a list of *m* documents ranked according to their semantic relevance with the input query. We must point again that in such an implementation, there is no use of any external resource.

The steps for computing the mapping between a free-form query and an SWD is depicted in Figure 5:

Figure 5 Computing the mapping of a free-form query to an SWD without external resources

1. Find the concepts C of each SWD that their lexicalisation matches the query terms of the free-form query.
2. For all C, get their associated concepts C_V (concepts in the vicinity) i.e. hyperonyms, hyponyms, equivalent, related concepts (with other relation).
3. For each SWD build a new document (SWD-New) that comprises the terms that lexicalise C and C_V, plus their label and comment properties values.
4. Build a query document for each term *t* in the query string using the rest of the terms in the query sting (i.e. *t* and its vicinity V_t), based on the VSM.
5. Find the ranked associations between the query document of term *t* and the SWD-New, by considering the association with the highest cosine coefficient similarity.

Experimenting with the Swoogle repository was rather hard to do in this case, due to numerous broken SWD's links (missing files) or non-proper file formats. For the SAMOS enhanced method, queries were placed against SWDs, using the 'def' prefix for each query term (e.g., def:play def:theater def:role) to retrieve SWDs that explicitly define the query term as a concept/property. We have also developed custom SWDs, i.e., OWL documents for specific domains of interest (booking flights, entertainment), in addition to related SWDs taken from Swoogle. Preliminary results obtained from experiments using these data resulted to precision results that are very close to the results obtained by the computation of the mapping of query terms to WordNet senses presented in Section 4. That is not surprising since both WordNet senses and SWDs are treated by the VSM in a similar way, i.e., as a set of documents. The advantages of the advanced SAMOS implementation over the basic one and other related approaches can be summarised in the following points:

- It speeds up the computations performed since
  - the step of query reformulation is missing
  - only a single mapping method is applied between the query and the SWDs, instead of the combination of mapping methods applied with AUTOMS ontology mapping tool
  - only a fragment (matched concepts and their vicinity) of the SWDs is considered in the mapping method instead of the large SWDs found in repositories such as Swoogle.
- There is no need for external resources, thus the approach can be applied independently from the existence and the efficiency of external resources.

Having said that, an important limitation of the SAMOS enhanced method, which we need to consider in our future work, is that SWDs must include rich information (related concepts, labels, comments) for the matched concepts in order for the VSM method to utilise a more representative semantic space.

As a general comment for the precision measure of all versions of the SAMOS system, we state that this is heavily dependent on the VSM performance, which we have not managed so far to raise over 80%. We conjecture that, as in any information retrieval case, precision could be further improved by sacrificing automation, i.e., by involving users at the early stages of query formulation. Furthermore, we envision a future implementation of SAMOS with the combination of advanced methods for the direct matching of queries with SWDs.

6 Conclusions

The problem of semantic search for heterogeneous and distributed SWDs has been addressed by recent approaches. Basically, a two-step approach is followed by all approaches:

- query reformulation (into triples or ontology)
- ontology matching.

This paper does not consider methodological issues towards solving this problem. Based on these related approaches, the paper presents an evaluated implementation of a two-step approach to SWDs' retrieval, namely SAMOS. However, going beyond the work of these approaches, the proposed method aims to support users to place free-form queries, requiring no additional knowledge and skills for expressing queries in a formal language. The paper also discusses important issues that must be considered towards approaching the retrieval of SWDs in such a manner. The presented implementation is evaluated using Swoogle's search engine by semantically re-ranking keyword-based results.

The paper placed much emphasis to the problem of automatically transforming free-form queries into formal ones, expressed as 'query-ontologies', by exploiting domain ontologies. The aim is to facilitate the semantic (as opposed to the specific case of keyword-based) retrieval of SWDs by matching the constructed query-ontology to the documents in a Semantic Web repository (using an automated ontology matching method) and ranking the retrieved documents according to their relevance to the query-ontology.

Targeting to advancing semantic search systems' implementations, the paper discussed two alternative implementations of the proposed system, SAMOS. Preliminary results have shown that it is possible to use a free-form query to semantically retrieve or re-rank SWDs without using a separate disambiguation stage or by constructing an explicit query-ontology. We have shown that there is no need for using heavy-duty automated ontology matching tools, but instead, there is evidence that a simple semantic mapping method for semantically matching query terms to SWDs suffices. We conjecture that further study of both approaches presented in this paper will result to an advanced solution of the semantic retrieval problem.

Acknowledgements

We are grateful for the valuable comments of the anonymous reviewers and the decisive management of Editor-in-Chief. We also thank our colleague Mr. Alexandros Valarakos for contributing during the implementation of the experimental system.

This work is part of research project ONTOSUM (www.ontosum.org), implemented within the framework of the "Reinforcement Program of Human Research Manpower" (PENED) and co-financed by EU-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

References

Alesso, P. (2004) 'Semantic search technology', *AIS SIGSEMIS Bulletin*, Vol. 1, No. 3, pp.86–98.

Beckett, D. (2004) *RDF/XML Syntax Specification (Revised), W3C Recommendation*, http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/, Latest version available at http://www.w3.org/TR/rdf-syntax-grammar/

Bernstein, A., Kaufmann, E. and Fuchs, N. (2005) 'Talking to the semantic web – a controlled english query interface for ontologies', *AIS SIGSEMIS Bulletin*, Vol. 2, No. 1, pp.42–47.

Bouquet, P., Kuper, G.M., Scoz, M. and Zanobini, S. (2004) 'Asking and answering semantic queries', Workshop on *Meaning Coordination and Negotiation Workshop (MCN-04) in Conjunction with the 3rd International Semantic Web Conference (ISWC-04)*, Hiroshima Japan.

Corcho, O. (2006) 'Ontology based document annotation: trends and open research problems', *Int. J. Metadata, Semantics and Ontologies*, Vol. 1, No. 1, pp.47–57.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. (1990) 'Indexing by latent semantic analysis', *Journal of the American Society of Information Science*, Vol. 41, No. 6, pp.391–407.

Ding, L., Finin, T., Joshi, A., Pan, R., Scott Cost, R., Sachs, J., Doshi, V., Reddivari, P. and Peng Y. (2004) 'Swoogle: A search and metadata engine for the semantic web', *Thirteenth ACM Conference on Information and Knowledge Management (CIKM'04)*, Washington.

Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., Hage, W. and Yatskevich, M. (2006) 'Results of the ontology alignment evaluation initiative 2006', *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006)*, *International Semantic Web Conference (ISWC-2006)*, Athens, Georgia, USA.

Fikes, R., Hayes, P. and Horrocks, I. (2003) *OWL-QL – a Language for Deductive Query Answering on the Semantic Web*, Technical Report, Knowledge Systems Laboratory, Stanford, CA.

Karanastasi, A. and Christodoulakis, S. (2007) 'Ontology-Driven semantic ranking for natural language disambiguation in the ontonl framework', The *Proceedings of the 4th European Semantic Web Conference (ESWC)*, 3–7 June, Innsbruck, Austria.

Karvounarakis, G. (2003) *The RDF Query Language (RQL)*, Technical Report, Institute of Computer Science, Foundation of Research Technology.

Kotis, K. and Vouros, G. (2006) 'Towards semantic web documents retrieval through ontology mapping: preliminary results', *1st Asian Semantic Web Conference (ASWC 2006) Workshop on Web Search Technology – from Search to Semantic Search*, Beijing, China.

Kotis, K., Vouros, G. and Stergiou, K. (2006a) 'Towards automatic merging of domain ontologies: the hcone-merge approach', *Elsevier's Journal of Web Semantics (JWS)* Vol. 4, No. 1, pp.60–79.

Kotis, K., Valarakos, A. and Vouros, G. (2006b) 'AUTOMS: automating ontology mapping through synthesis of methods', *OAEI (Ontology Alignment Evaluation Initiative) 2006 Contest, Ontology Matching International Workshop*, Atlanta USA. Also in CEUR-WS.org/Vpl. 225, ISSN 1613-0073.

Lopez, V., Motta, E. and Pasin, M. (2005) 'AquaLog: an ontology portable question answering interface for the semantic web', *European Semantic Web Conference 2005*, Heraklion, Crete.

Lopez, V., Motta, E. and Uren, V. (2006a) 'PowerAqua:fishing the semantic web', *European Semantic Web Conference 2006*, Montenegro.

Lopez, V., Sabou, M. and Motta, E. (2006b) 'PowerMap: mapping the real semantic web on the fly', *International Semantic Web Conference*, Georgia, Atlanta.

Miller, G. (1995) 'WordNet: A lexical database for English', *Communications of the ACM*, Vol. 38, No. 11, pp.39–41.

Patel-Schneider, P.F., Hayes, P. and Horrocks, I. (Eds.) (2004) *OWL Web Ontology Language Semantics and Abstract Syntax*, *W3C Recommendation*, 10 February, Latest version is available at http://www.w3.org/TR/owl-semantics/

Raghavan, V.V. and Wong, S.K.M. (1986) 'A critical analysis of vector space model for information retrieval', *JASIS*, Vol. 37, No. 5, pp.279–287.

Salton, G. and McGill, M.H. (1983) *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, NY.

Straccia, U. and Troncy, R. (2006) 'Towards distributed information retrieval in the semantic web: query reformulation using the omap framework', *Proceedings of the 3rd European Semantic Web Conference (ESWC 06)*, pp.378–392.

Tamma, V., Blacoe, I., Lithgow-Smith, B. and Wooldridge, M. (2004) 'SERSE: searching for semantic web content', *Proceedings of the Sixteenth European Conference on Artificial Intelligence (ECAI-04)*, Valencia, Spain.

Valarakos, A., Spiliopoulos, V., Kotis, K. and Vouros, G. (2007) 'AUTOMS-F: a java framework for synthesizing ontology mapping methods', Accepted for publication in *I-KNOW 2007 Special Track on Knowledge Organization and Semantic Technologies 2007 (KOST '07)*, 5 September, Graz.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H. and Studer, R. (2006) 'Semantic wikipedia', *WWW 2006*, 23–26 May, Edinburgh, Scotland.

Zhang, L., Yu, Y. and Yang, Y. (2005) 'An enhanced model for searching in semantic portals', *WWW 2005*, 10–14 May, Chiba, Japan.

Websites

[1]http://www.yahoo.com

[2]http://www.w3.org/TR/owl-ref/

[3]http://www.ontosearch.org/

[4]http://www.google.com/technology/

[5]http://www.ask.com/