An Approach to Semantic Information Retrieval Based on Natural Language Query Understanding

Beniamino Di Martino

Second University of Naples
Dipartimento di Ingegneria dell' Informazione
beniamino.dimartino@unina.it

Abstract. In this paper we present an approach to semantic based Information Retrieval of semantically annotated documents, based on natural language understanding of the query, on its semantic representation through an OWL query ontology, and on exact and approximate retrieval of semantically annotated documents, through SPARQL inference and structural graph matching.

1 Introduction

Semantic Web is increasingly gaining momentum as a viable model (and accompanying technology) to represent and manage content and knowledge on the Web at the semantic level. One of the web activities and technology that Semantic Web promises to deeply renovate is Information Retrieval, expecially when documents and web sites and resources (services) are described at the semantic level, with use of OWL-based *semantic annotations* [1,14].

In order to be able to express their information need, users of current Semantic Web technology and tools need to be familiar with Ontology languages' syntax (such as RDF and OWL), with formal query languages (such as RDQL and SPARQL), and with schema and dictionaries of target ontologies. In order to overcome such a gap, interfaces based on Natural Language Processing (NLI – *Natural Language Interfaces*) might be effective in expressing the information need, and thus in querying ontologies and semantic annotations in an intuitive and familiar mean, hiding to the non-expert user the complexity of formal grammars of ontologies, semantic annotations and query languages.

A number of problems need to be faced when addressing such approach; they include: the complexity and intrinsic ambiguity of the natural language, which limits the viability of grammar based parsers and stocastical parsers as well; the difficulty in translating the user query into a formal query expressed in formal languages such as SPARQL; the need to map terms expressing concepts and relations used in the user query with corresponding components of, possibly standard, domain ontologies; the need to map the user query with semantic based descriptions of the documents and web pages: semantic annotation models and tools are under development and still not mature enough.

In this paper we present an approach to semantic based Information Retrieval of semantically annotated documents and web sites, and a prototypical tool, which tries to tackle these problems. It is based on natural language query processing at syntactical level, from which semantic patterns are heuristically determined, and are matched (using graph based matching algorithms) against domain ontologies, in order to extract an ontology based representation of the user query (*query ontology*). It is then,

F. Daniel and F.M. Facca (Eds.): ICWE 2010 Workshops, LNCS 6385, pp. 211–222, 2010.
© Springer-Verlag Berlin Heidelberg 2010

after possible refinement from the user, either translated into SPARQL query in order to perform an exact retrieval of documents annotated with the instances found, either matched against the semantic annotations which represents the documents' meaning, in order to perform an approximate retrieval, ranked with matching measures.

The paper is structured as follows: section 2 provides with a brief overview of methods and approaches to semantic based information retrieval and semantic annotation; section 3 describes the defined technique and developed architecture, section 4 illustrates the implemented prototype tool, *Semantic Searcher*, through an example.

2 Background and State of the Art

Semantic based models and systems for Information Retrieval try to augment (or in some cases to replace) traditional IR technology with Knowledge Engineering technology, in order to solve or at least improve the classical trade-off "precision vs recall" which traditional IR methods suffer [2]. Several approaches to IR based on semantics exist, including *profile-based* approaches (where user preferences and interest domains are collected and represented, and are measured for similarity with suitable representations of document content, see e.g. [3]), *collaborative* approaches (where similarity is measured among users, instead than document content, see e.g. [4]), and *classification* approaches (where documents are classified, automatically or manually, according to predefined categories/taxonomies, or are unsupervisedly clustered, see e.g. [5,6]). The approaches based on Ontologies relies on standardized machine-readable representations of knowledge, domain oriented but also *upper-level*, in standard and open languages like RDF and OWL. Main efforts and systems include (among others): OntoSeek [7], On2Broker [8], WebSifter [9], Score [10], Intellizap [11], Swoogle [12]. We cannot provide details on these systems here for reasons of space. In [13] a detailed overview and comparison of those systems is provided. Ontology-based annotation of documents and media is one of the most active research topics in making documents machine understandable and allowing for effective document information retrieval. First examples of documents annotated using metadata can be found in the (*KA*)2 initiative [14]. In [15] a detailed state of-the-art overview is provided and a novel annotation model and a prototype tool is proposed.

3 A Technique and an Architecture for Semantic Information Retrieval

The technique we have devised is composed by the following phases:

1. Syntactical analysis of the query, expressed in Natural Language
2. Building from the syntactic tree of a Query graph (by means of Triple extraction)
3. Mapping of Query graph with domain ontologies
4. Query ontology graph production and user validation
5. Semantic querying with (alternatively):
    a. Ontology query translation in a semantic query language (e.g. SPARQL) and running of a SPARQL engine (such as Jena ARP or Pellet)
    b. Graph matching of query ontology with semantic annotations
6. Visualization of results

The defined steps are visually shown in figure 1.

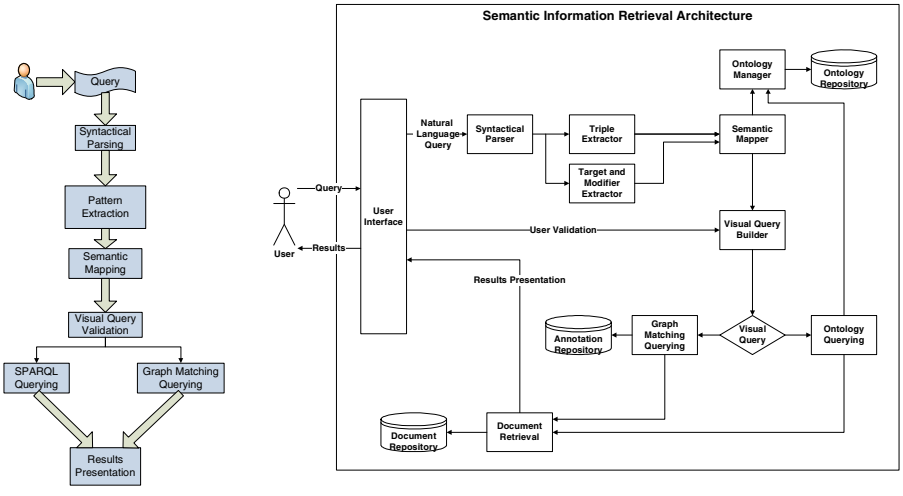

Fig. 1. The Semantic Information Retrieval procedure and Architecture

The above defined technique is detailed by means of the architecture depicted in figure 1. It consists of the following components:

- ***User Interface*:** it allows the user to input the query phrase (in natural language) and to drive her/him through the following phases.
- ***Ontology Manager*:** it manages the local ontologies and semantic annotations, by ensuring consistency and persistence.
- ***Syntactical Parser*:** it performs lexical and syntactic analysis of the user query.
- ***Pattern Extractor*:** such component extracts from the syntactic tree a skeleton composed by all syntagms found in the query, detects the noun syntagms and then detects all the relationships among them (represented by verb, adjective and other noun syntagms), thus extracting a set of patterns in the form of triples.
- ***Semantic Mapper*:** such component maps the patterns extracted from the query with components of the domain ontologies through graph matching algorithms.
- ***Visual Query Builder*:** this component presents the extracted query ontology to the user, who can possibly refine it, by browsing the domain ontologies from which the query ontology has been extracted, select other concepts and/or relationships from them, and add to the query ontology or replace components.
- ***Semantic Querying*:** this component performs the translation of the produced query ontology into a SPARQL query, and executes it, finding all the instances which fulfill the semantic query.
- ***Graph Matching Querying*:** this component (alternative to the previous one) performs the search in a different way, by doing a graph matching among the query ontology graph, and all the semantic annotation graphs managed by the ontology manager.
- ***Document Retrieval and Presentation*:** this component present the user with the results of the SPARQL querying or the graph matching querying, that is the

found instances, the annotations which present those instances as a component, and the annotated documents.

In the following ssubsections we illustrate in more details the functionalities of some of the main components of the architecture.

3.1 Ontology Manager

The *Ontology Manager* component manages the local domain ontologies and representations of semantic annotations. It maintains their persistence and versioning with respect to multiple accesses and modifications. It indexes the terms used to define concepts, instances and relations, in order to allow for a faster retrieval of the domain ontologies relevant to the user query, when the Semantic Mapping is performed. Ontologies, represented in OWL language, are parserized and then the terms indexed after stopwords removal and stemming.

3.2 Syntactical Parser

This component parses the natural language query, performing lexical and syntactical analysis, and outputs the Syntax tree of the query phrase. We have implemented a Context Free Grammar for the Italian language, making use of Prolog and its feature to easily implement CFGs through built-in Definite Clause Grammars.

3.3 Pattern Extractor

This component extracts from the Syntax tree heuristic patterns, to be then matched against domain ontologies.

The patterns we have defined are RDF-like triples, composed from syntax tree syntagms, of the form <*subject, predicate, object*>, where *subject* and *object* are composed from noun syntagms (in the following named as NS), while *predicate* can composed by one or multiple Verb syntagms (VS), prepositional syntagms (PS), adjective syntagms (AS), conjunction syntagms (CS) and noun syntagms (NS) as well, or can be a predefined OWL relationship. The reason for choosing such triples as patterns is because OWL has as main component the RDF triples, which represent relationships (the predicate) between two classes or class instances (the subject and object). Thus matching patterns against the domain ontologies is equivalent to find subgraph isomorphisms between the pattern triples and subgraphs triples of the ontologies.

The pattern triples are identified from typical aggregations of syntagms in the syntax tree, with optional constraints. The most typical syntagms are of the form:

NS-VS-NS
NS-PS-NS
NS-NS-NS
NS-AS-NS

of which the first is the most frequent, but also:

SN-(SV,SN)-SN

or SN alone, composed by an adjective and a noun.

For instance, the pattern NS-VS-NS with VS an auxiliary verb (e.g. the italian verbs *essere* (to be) or *avere* (to have)) can be mapped to a pattern RDF triple where the predicate part is the predefined OWL relationship *subClassOf*, as in the example:

La birra <NS> è <VS> una bevanda <NS>

(beer is a drink)

which is mapped to the RDF triple shown in figure 2.

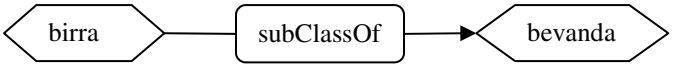

Fig. 2. An RDF triple pattern

Another example is the same pattern NS-VS-NS with VS auxiliary verb and the first NS containing a proper noun. It can be mapped to the RDF triple where the predicate part is the predefined OWL *instanceOf* relationship, as in the example:

La Lager <NS> è <VS> una birra <NS>

(Lager is a beer)

which is mapped to the RDF triple shown in figure 3.

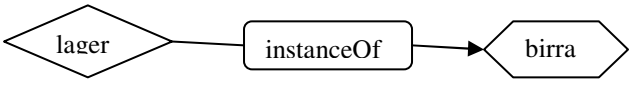

Fig. 3. Another RDF triple pattern

The pattern NS alone, composed by a noun and a qualificative adjective, can be mapped to two different RDF triple patterns; infact the adjective can be a specifier or an attribute of a noun. In the first case, the predicate of the RDF triple can be the *subClassOf* OWL relationship, while in the other case it can be any relationship or the *instanceOf* relationship, thus in the second case the predicate part is left undefined, and it will be "filled" when a match is found with a corresponding triple of the ontology graph by the graph matching algorithm. As an example:

La birra rossa <NS>

(the red beer)

is mapped to the two RDF triples shown in figure 4.

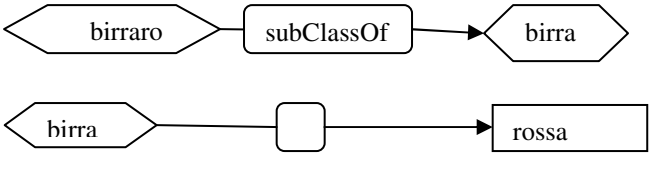

Fig. 4. Two RDF triples from <NS>

The pattern NS-VS-NS-NS where the VS is an auxiliary verb, is mapped to an RDF triple where the predicate part is comosed by concatenation of the middle pair (VS,NS), as in the example:

La birra <NS> ha <VS> come ingrediente <NS> il malto <NS>
(beer has malt as ingredient)

is mapped to the RDF triple shown in figure 5.

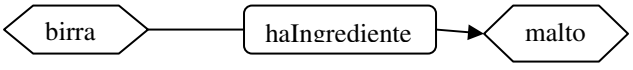

Fig. 5. An RDF triple from NS-VS-NS-NS

As a final example, the pattern NS-PS-NS where the prepositional syntagm PS is the italian preposition *di* and the second NS is composed by a noun and an adjective, can be mapped to two RDF triple patterns; infact the preposition *di* coupled with a noun can have a meaning of a specifier or an attribute. In the first case, the predicate of the RDF triple can be the *subClassOf* OWL relationship, while in the second case it can be the *instanceOf* relationship, as in the example:

La birra <NS> di <PS> colore rosso <NS>
(the beer of red color)

Is mapped to the two RDF triples shown in figure 6.

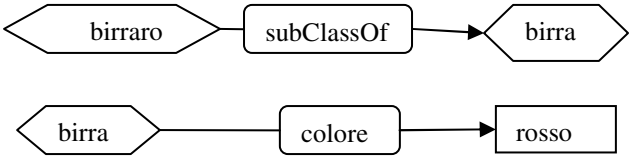

Fig. 6. Two RDF triples from NS-PS-NS

3.4 Semantic Mapper

This component tries to match the pattern triples found by the Pattern Extractor with the components of the domain ontologies stored in the Ontology manager.

First of all, the relevant ontologies are detected by traditional search of terms present in the query within the ontologies' indices, possibly expanded by their synonyms (we use Multiwordnet [16] thesaurus at this purpose). Each of these ontologies is then parsed and a graph representation for each of them is obtained (OG – Ontology Graph). The set of triples detected by the Pattern extractor is also represented as a single graph (QG – Query graph). Then a structural matching algorithm, based on subgraph isomorphism at the structural level and on string similarity and use of synonyms at the node - link level, is applied to each couple (OG, QG). The details of the matching algorithm and the developed procedure cannot be illustrated here for reasons of space, but can be found in [17]. The subgraphs matched, weighted with a similarity measure, are returned to the user, through the following component, the Visual Query Builder, in form of a *Query Ontology*.

3.5 Visual Query Builder

This component visualizes the matched subgraphs to the user, in the form of a *Query Ontology*, which represents the meaning of the natural language query, together with the syntactic tree and the matched domain ontology(ies). The user can possibly refine it, by browsing the domain ontologies, select other concepts and/or relationships from them, and add to the query ontology or replace components of it. The resulting refined query ontology is then passed to the Semantic Querying component.

3.6 Semantic Querying

This component performs the actual retrieval of semantically annotated documents, by means of the constraints defined with the Query Ontology automatically produced from the natural language query, possibly refined by the user. The retrieval can be performed in two ways:

- with an *exact* approach, based on translation of the ontology into a SPARQL query, which, executed by an engine, returns the instances which exactly fulfill the constraints of the query, and subsequently the semantic annotations using those instances and the linked documents;
- with an *approximate* approach, based on graph matching of the ontology query with the semantic annotations linked to the documents. The matches are weighted and thus the documents corresponding to the matched annotations can be scored in order of similarity with respect to the query ontology.

The translation of the query ontology into a SPARQL query cannot be illustrated here for space limitations; please refer to [18] for details.

4 Semantic Searcher: A Prototype Tool for Semantic IR

We have implemented the procedure and architecture defined in the previous section, and the result is the prototype tool ***Semantic Searcher***. It has been developed by using the Java development environment Eclipse, by utilizing Jena [19] parsers for OWL and SPARQL, Jena SPARQL engine ARQ, and reasoner Pellet [20]. The domain ontologies' indexing has been implemented by utilizing the Information Retrieval library *Lucene*. Multiwordnet [16] synonyms thesaurus has been utilized for expanding the graph matching and the ontologies' indexing. Natural Language parsing and conceptual graph representation has been performed (for the Italian Language) with use of SWI Prolog interpreter. We present in the following the tool's functionality by means of an example query process, performed in Italian language (we provide anyway English translation of all the phases of the querying procedure).

The chosen example domain is *beers*. As an example user wants to know which beers are made with a given ingredient and have a given colour. Then she/he expresses the following query: "Which beers of red colour have malt of wheat as ingredient?" (in Italian: "*Quali sono le birre di colore rosso che hanno come ingrediente il malto di frumento?*"). The natural language query is initially analized by the *Syntactical Parser* which outputs the syntactical tree (it can be visualized in figure 8). The parse tree is then inputted to the *Semantic Mapper* which performs the mapping to the domain ontologies available to the *Ontology Manager.* Once performed the mapping,

the tool returns (in the *Result Matching* panel) the matched subgraphs, together with quantitative results of the graph matching, as shown in figure 7.

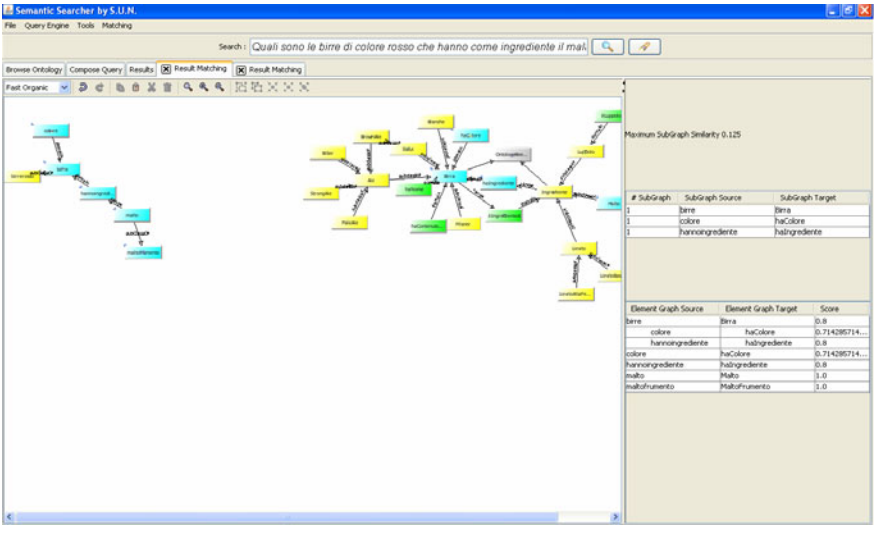

Fig. 7. Results of the matching phase

The best matched subgraph is then translated into a *Query Ontology*, which represents the meaning of the natural language query, and presented to the user, together with the syntactic tree and the matched domain ontology(ies) (in the *Browse Ontology* panel) as shown in figure 8, together with the query syntax tree.

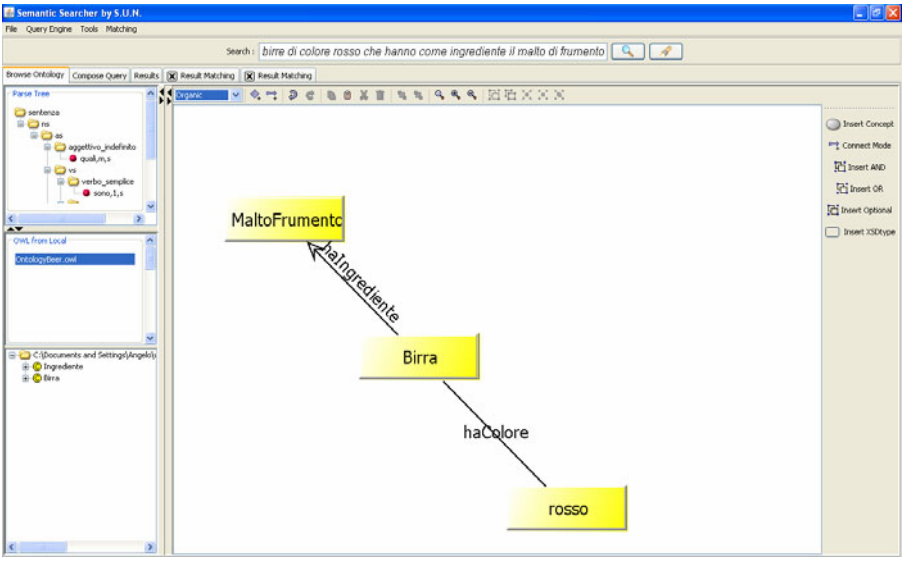

Fig. 8. The extracted query ontology, representing the meaning of the natural language query

As shown in the figure, the query ontology extracted form the example phrase is composed by three concepts - *Birra* (beer), *MaltoFrumento* (WheatMalt) and *Rosso* (red) - linked by the relations *haIngrediente* (has ingredient) and *haColore* (has color), which are component of the domain ontology (OntologyBeer.owl) which has matched, as shown in left panel of figure 8.

Once visualised the Query ontology, the user can possibly refine it, by browsing the domain ontologies from which the query ontology has been extracted, select other concepts and/or relationships from them, and add to the query ontology or replace components by means of "drag and drop" action, as shown in figure 9.

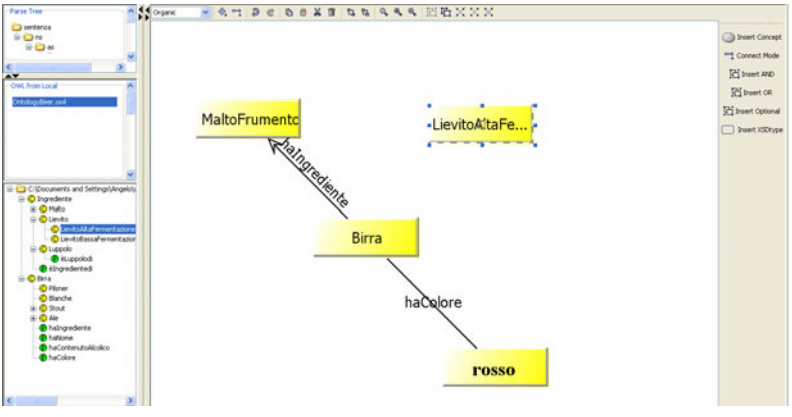

Fig. 9. "Drag and drop" from the domain ontology to the query ontology

Once completed the refinement of the query ontology, it is transferred to the *Compose Query* panel, where the semantic querying is performed in order to find the relevant semantically annotated documents. The user can choose to perform an (exact) semantic querying with a SPARQL engine – she/he can select the kind of semantic engine used to perform the semantic querying (the Jena SPARQL engine ARQ or reasoner Pellet) – or to perform a structural matching between the query ontology and the semantic annotations, with the same algorithms used in the previous phase.

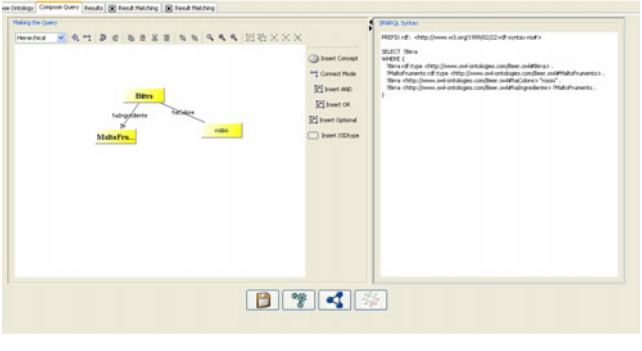

Fig. 10. Generated SPARQL query

In figure 10 it is shown the generated query in SPARQL language (which can also be manually modified), while figure 11 shows the returned instances found.

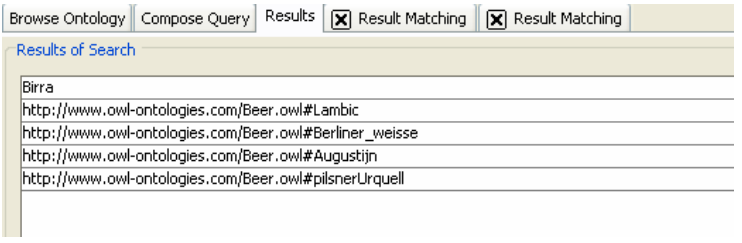

Fig. 11. Instances found from the semantic engine

The instances found (*Lambic, Berliner_weisse, Augustin and PilsenerUrquell*) satisfy the query constraint: they're red beers with WheatMalt as one of the ingredients. The search is *exact*, thus there is no need for ranking.

By clicking on an instance, the corresponding semantic annotations which have used that instance can be visualized, as shown in figure 12, and by clicking on an annotation, the corresponding annotated document is visualized, as shown in fig. 13.

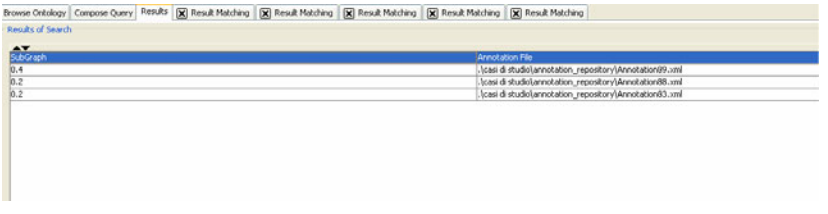

Fig. 12. Annotations containing a selected found instance

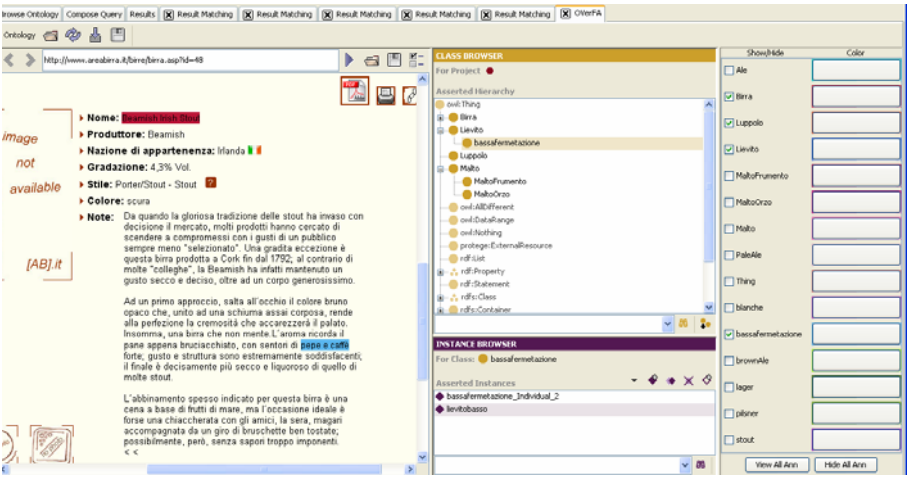

Fig. 13. The annotated document

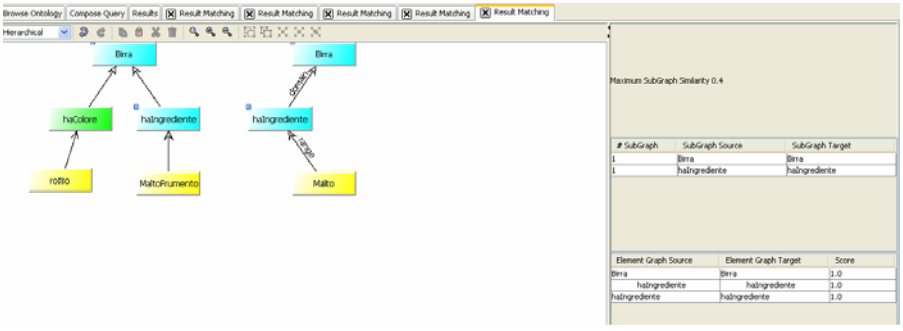

Fig. 14. Results of Graph Matching between query and an annotation

The alternative to the SPARQL semantic search is the search of annotations by graph matching with the query ontology. Figure 14 shows the results of matching the example query ontology graph with a semantic annotation graph, which represents a given annotation.

As shown in figure 14, the match (which is not exact since the applied algorithm is subgraph isomorphism) is measured with the similarity measure defined for subgraph isomorphism algorithm. This measure can be used to rank the results found (currently each matched annotation is shown on a separate panel, as seen in figure 14, in order to browse and inspect each of the graph matches).

5 Conclusions

We have presented a technique, an architecture and a prototypical implementation for Information Retrieval based on natural language understanding of query, on representation of semantics of the user need and of the documents by means of OWL ontologies and semantic document annotations, and on structural graph matching. It shows that an exact match between user need and semantics of document can be obtained, if documents are semantically annotated, with ontological representation of user need, obtained automatically with NLP and validated by the user, and with use of query language SPARQL and inference engine Pellet. Inexact matching can be obtained instead with graph matching of ontology based annotations and representation of user query. Similarity measures can be defined for the inexact graph matching, and used to rank the search results. Definition of such a measure, and validation with experimental campaign and analysis is subject of future work.

Acknowledgements

The work described has been supported by the PRIST project “Fruizione assitita e context aware di siti archeologici complessi mediante dispositivi mobili”. I would like to thank Angelo Martone for the prototype implementation.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic web. Sci. Am. 1(1), 68–88 (2000)
2. Baeza-Yates, R., Navarro, G.: Integrating contents and structure in text retrieval. SIGMOD Rec. 25(1) (1996)
3. Ackerman, M., et al.: Learning Probabilistic User Profiles - Applications for Finding Interesting Web Sites, Notifying Users of Relevant Changes to Web Pages, and Locating Grant Opportunities. AI Magazine 18(2), 47–56 (1997)
4. Bollacker, K.D., et al.: Discovering Relevant Scientific Literature on the Web. IEEE Intelligent Systems 15(2), 42–47 (2000)
5. Koshman, S., Spink, A., Jansen, B.J.: Web Searching on the Vivisimo Search Engine. Journal of the American Society For Information Science And Technology 57(14), 1875–1887 (2006)
6. Labrou, Y., Finin, T.: Yahoo! as an ontology: using Yahoo! categories to describe documents. In: Procs of the eighth intl. conference on Information and knowledge management, Kansas City, USA, pp. 180–187 (1999)
7. Guarino, N., et al.: OntoSeek: Content-based Access to the Web. IEEE Intelligent Systems 14(3), 70–80 (1999)
8. Fensel, D., et al.: On2broker: Semantic-Based Access to Information Sources at the WWW. In: Proceedings of the World Conference on the WWW and Internet (WebNet 1999), Honolulu, Hawaii, USA, pp. 25–30 (1999)
9. Kerschberg, L., Kim, W., Scime, A.: WebSifter II: a personalizable meta-search agent based on weighted semantic taxonomy tree. In: Proc. of Int. Conf. on Internet Computing (2001)
10. Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Managing Semantic Content for the Web. IEEE Internet Computing 6(4) (2002)
11. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing Search in Context The Concept Revisited. ACM Transaction on Information Systems 20(1), 116–131 (2002)
12. Finin, T., Ding, L., Pan, R., Joshi, A., Kolari, P., Java, A.: Swoogle: Searching for knowledge on the Semantic Web. In: Proc. of Intl. Conf. on Artificial Intelligence (2005)
13. Di Martino, B., et al.: Stato dell' arte ed analisi delle tecnologie e degli strumenti per l'Information Retrieval. Technical Report TR2.4.4, Progetto LC3 (2009)
14. Benjamins, V.R., Fensel, D., Decker, S., Gomez-Perez, A. (KA)2: building ontologies for the internet: a mid term report. Intl. J. of Human Computer Studies 51, 687–712 (1999)
15. Moscato, F., Di Martino, B., Venticinque, S., Martone, A.: OverFA: A collaborative Framework for Semantic Annotation of Documents and Web Sites. International Journal of Web and Grid Services (IJWGS) 5(1), 30–45 (2009)
16. Pianta, E., Bentivogli, L., Girardi, C.: MultiWordNet: Developing and Aligned Multilingual Database. In: Proceedings of the First International Conference on Global WordNet, Mysore, India, January 21-25, pp. 293–302 (2002)
17. Di Martino, B.: Semantic Web Services Discovery based on Structural Ontology Matching. International Journal of Web and Grid Services (IJWGS) 5(1), 46–65 (2009)
18. Di Martino, B., et al.: Definizione di una Architettura per il Semantic Information Retrieval, Technical Report TR2.4.9, Progetto LC3 (2009)
19. Jena – A Semantic Web Framework for Java, `http://jena.sourceforge.net/`
20. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner, Tech. Rep. CS 4766, University of Maryland, College Park (2005)