Ontology Driven Semantic Search

DARIO BONINO, FULVIO CORNO, LAURA FARINETTI, ALESSIO BOSCA
Dipartimento di Automatica ed Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 10129 Torino
ITALY
{dario.bonino, fulvio.corno, laura.farinetti, alessio.bosca}@polito.it

Abstract: - The introduction of semantics on the web will lead to a new generation of services based on content rather than on syntax. Search engines will provide topic-based searches, retrieving resources conceptually related to the user informational need. Queries will be expressed in several ways, and will be mapped on the semantic level defining topics that must be retrieved from the web.

Moving towards this new Web era, effective semantic search engines will provide means for successful searches avoiding the heavy burden experimented by users in a classical query-string based search task. In this paper we propose a search engine based on web resource semantics. Resources to be retrieved are semantically annotated using an existing open semantic elaboration platform and an ontology is used to describe the knowledge domain into which perform queries. Ontology navigation provides semantic level reasoning in order to retrieve meaningful resources with respect to a given information request.

Keywords: - Semantic retrieval, Ontology navigation, Semantic annotation, Query refinement, DOSE.

1. Introduction

Despite the great improvement in search effectiveness of nowadays search engines, failures on capturing the information request semantics are still not addressed. The classical query string interface does not provide means for the identification of relevant concepts that must be contained into retrieved resources, and query terms are still used as syntactic descriptors for the page content.

Although modern engines index the whole content of web pages, the search task is less or more built on term matching between the user query and the engine database, enriched by some fine Information Retrieval techniques ranging from the *tf*/*idf* ranking model [2] to the famous Google PageRank [3].

Those techniques form the "state of art" of IR at the syntactic level, however a great enhancement in result relevance could be achieved by bridging the gap between syntax and semantics. Knowing "exactly" what the user means when specifying a search query, and having content descriptions of web resources would allow retrieval systems to provide focused results, and to better satisfy users.

The introduction of such new generation search engines will not compete with existing technologies and instead of replacing them, a more powerful integration between the semantic and the syntactic level will be adopted, promoting the availability of text and topic wise search services.

Many recent researches provide rich architectures for semantic support on the Web, the KAON system [1] as an example provides the infrastructure for building ontology-based portals targeted to business applications. The authors of this paper developed an open semantic web platform [4] which allows automatic semantic annotation of web resources at the document substructure level and provides basic annotation-based search functionalities. Several research projects share the goal of semantics introduction on the web, the On-To-Knowledge [5] as an example or the European Socrates/Minerva CABLE project [6]that provides a case-based e-learning infrastructure for educators.

In this paper we propose a search engine based on ontology navigation able to use semantic annotations about web resources in order to provide relevant results. The engine intelligence consists of an automatic search relevance detection mechanisms that is able to trigger appropriate navigation on the concept hierarchy defined by a domain ontology. The retrieval model is able to infer the required level of detail for a given query and is "trainable" to fine-tune precision and recall performance.

The paper organization is as follows: in the second section an overview of the *tf*/*idf* based vector model is provided and the application of a vector-model like retrieval mechanism at the concept level is discussed. Section 3 describes ontology navigation while section 4 describes the intelligent relevance detection mecha-

nism proposed in this paper for improving relevance of search results.

In section 5 a preliminary implementation of the proposed search engine onto the DOSE platform is described while section 7 shows some experimental results. Finally section 8 draws conclusions and proposes some future works.

2. Conceptual IR Model

2.1 Classical vector space model

The classical IR vector model was originally proposed as an alternative to the boolean model in order to overcome its limitations in terms of crisp definition of document relevance. The vector model is based on the specification of a set of keywords that compose an orthogonal base for a multidimensional vector space. Each document indexed using this model possesses a relevance vector expressed in that space. Vector components are defined as the relevance of the represented document with respect to a specific keyword, according to a given weighting scheme, the *tf/idf* [2] as an example.

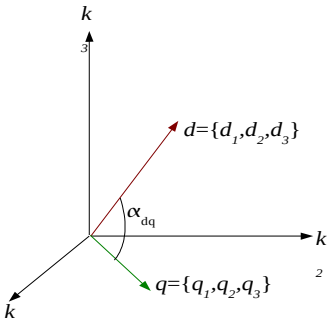

Fig. 1 Vector Space Model

In the vector model a document *d* and a user query *q* are represented as *n*-dimensional vectors where *n* is the number of keywords. The degree of similarity of the document *d* with regard to the query *q*, and thus the relevance ranking, is computed as the cosine of the angle between the vectors *d* and *q*.

$$Simil(d, j) = \cos(\alpha_{dq}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}||\vec{q}|} \quad (1)$$

The IR problem can be viewed as a clustering problem, determining which objects are in the set *A* specified by the user query and which not. First one needs to determine which are the features that better describe the set *A*, and secondly which features better distinguish resources in the set *A* from the others. The first fact is taken into account by the term frequency (*tf)* factor and provides a measure of how well a term describes the document content, while the second is accounted into the inverse document frequency (*idf)* factor which quantifies how much common is a term in the document set.

Def: *Let N be the total number of documents in the system and n_i be the number of documents in which the index term k_i appears. Let $freq_{i,j}$ be the raw frequency of term k_i in the document d_j. Then the normalized frequency tf_{ij} of that term in d_j is*

$$tf_{ij} = \frac{freq_{ij}}{max_l(freq_{lj})} \quad (2)$$

Where the maximum is computed over all terms mentioned in the document d_j. Now, let idf_i, inverse document frequency for k_i, be given by

$$idf_i = \log\left(\frac{N}{n_i}\right) \quad (3)$$

The best known weighting scheme based on preceding values is given by

$$w_{ij} = tf_{ij} \times idf_i \quad (4)$$

Where w_{ij} correspond to the i-th component of the document d_j vector representation.

2.2 Concept level vector model

In this paper, the information retrieval task is approached at an higher level of abstraction with respect to the classical scenario. In fact, in a semantic search platform, retrieval should act at the concept level and work on semantic annotations in order to provide relevant results in response to conceptual queries.

The classical vector model is still useful and can effectively be adopted by properly re-defining the components involved into the model such as keywords, documents and queries. At the concept level no syntactic entities like relevant terms exist, however the role played by those terms into the IR model can be assumed by concepts coming from a domain ontology. Therefore documents at the semantic level could be represented as vectors located in an hyperspace defined by the set of all ontology concepts, we named this representation "*conceptual vector*" .

Documents are still web resources and they are semantically described by annotations composed of RDF triples which characterize a conceptual link between a resource and an ontology node (Fig.2) .

The basic components of an annotation are an XPointer for resource identification, a weight that plays the role of w_{ij} in the classical vector space model and a concept identifier. Given a query *q,* i.e. a *conceptual vector* which represents user informational need, and a web resource *d* the

similarity between the query and the resource is defined exactly as in (1).

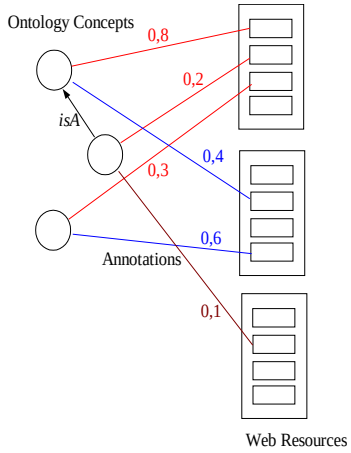

Fig. 2 Example of resource annotation

Application of the vector space model takes advantage from the coherence between query and document representation; without regarding the process of information gathering from users, final queries must thus be expressed as vectors in the ontology hyperspace. That is to say, queries will be represented as sets of weighted concepts, where the weight of each component defines how much the user is interested in the corresponding concept modeled by the ontology.

There are at least two important issues to keep evidence of when applying the vector model at the conceptual level: the first one is related to the query format, in fact, as it is unlikely that users will specify weighted queries, useful interfaces have to be developed in order to capture user informational needs and convert them to the format used by the IR system. The second is related to the basic assumptions onto which the vector space model is based.

Such model assumes that keywords form an orthogonal base for the space of documents and query. Ontology concepts are not independent entities, but semantics rich ones, inter-related by different kinds of relationships, at least inheritance. Concepts related by means of a semantic relation, hierarchy as an example, are clearly dependent and cannot compose an orthogonal base. To better clarify this statement think about a general concept as “mean of transport”, both “car” and “train” are examples of child concepts. Those three concepts are clearly not independent, i.e., they compose a non-orthogonal base.

In order to take in to account this peculiarity of the semantic representation, we developed a query refinement process, based on ontology navigation, which use semantic relationships to find co-related terms for the query, and that is powered by the notion of semantics that is associated with the ontology model.

3. ONTOLOGY NAVIGATION

The key points for a semantic search refinement process are the availability of a domain ontology and the ability to understand semantic relationships between ontology concepts.

There are many relationships that can be effectively used to perform query expansion and term re-weighting. In fact, as known in the Semantic Web, semantics is captured into ontologies by means of generic relationships and it is the ability to correctly interpret the meaning of such relations, together with the capability to compute logic inference on them, that makes the new generation of the web so powerful.

The simplest relation which always appears into ontologies is inheritance, that relates concepts by specifying which ones are sub-concepts of more general ones. The inheritance relation usually defines the taxonomic structure of an ontology organizing concepts into trees where child nodes are connected to parent nodes by means of *isA* relationships (inheritance).

Focalization and generalization are the semantic operators that allow taxonomy navigation. Focalization is formally defined as follows:

Def: *Let c be a concept into a domain ontology D, a focalization step is an operator F such that*

$$F(c)=\{c_i | c_i = isA(c)\} \quad (8)$$

Generally speaking, a focalization step moves the focus from a given concept *c* to the child nodes of the same concept, thus providing a more detailed view of the knowledge modeled by *c*.

The generalization step is the operator that works in opposition to the focalization one. Starting from a given concept *c*, the generalization operator provides the parents of that ontology node (more than one if multiple inheritance is allowed). Formally:

Def: *Let c be a concept into a domain ontology D, a generalization step is an operator G such that*

$$G(c)=\{c_i | c = isA(c_i)\} \quad (9)$$

Both operators have a one-to-many cardinality, i.e., they provide more than one concept starting from a single one. While the generalization can be used "as is" because parent concepts represent different views of the child concept, the focalization step requires a decision process to guide the navigation of the ontology toward the area which corresponds to the user interest.

In a taxonomic ontology (an ontology in which only *isA* relationships are provided) focalization and generalization are the only operators needed in order to refine queries using semantic information, and this is the case that will be presented in the next sections.

However, if more general relationships are defined in the ontology, and if one wants to integrate the information captured by them into the query refinement process, more complex operators based on ontology inference are needed.

4. Intelligent Semantic Search

The main reason for applying the *tf/idf* vector space model at the semantic level is the possibility of refining the query terms (concepts and weights) using knowledge about concept relationships, i.e. semantics. In classical IR the query refinement process basically consists of two complementary steps: query expansion and term re-weighting. In the first step an original query which may be rather vague is focalized towards user needs by adding new terms to the query itself, while in the second step already available query terms are re-weighted in order to move the new query towards relevant resources and away from not relevant ones. The first step may sometimes involve the user into a so-called relevance feedback, but is usually based on correlated term finding, while the second strongly involves interaction with users or user models, requiring them to point out which retrieved resources are good and which not.

Query expansion and term re-weighting could be done, in a semantic framework, by taking into account semantic relationships between concepts; in that case there is no need of correlated term finding as concepts are already related to each other and it is possible to leverage the ontology structure for choosing new query terms. Working with ontologies allows to set up semantic level relevance feedback with users by making available interfaces for query concept focalization, generalization, etc. and can also allow the development of some intelligent agents able to judge the result set relevance with respect to the query and which can autonomously decide to focalize or widen the query in order to provide valuable results.

4.1 Approach overview

In this paper we propose an automatic search refinement mechanism based on ontology navigation; only taxonomic relationships are taken into account, therefore focalization and generalization are sufficient to compute semantic query refinement.

The approach works as follows: when an application level object requires a retrieval, it must specify three parameters, the relevance threshold, the number of relevant documents to be retrieved and the conceptual query. The conceptual query is composed by a sequence of concepts and related weights and is submitted to a vector space based retrieval system which provides a set of annotations pointing at relevant resources, with associated relevance values measured as the cosine of the angle between the resource vectors and the query vector. Results are ranked by relevance.

The threshold on relevance specified by the external application, that uses the search system, discriminates relevant documents from non relevant ones.

Ultimate goal of the search engine is to provide, at least, a set of relevant documents as wide as specified by the calling application. To achieve this goal it uses a query expansion technique powered by ontology navigation.

A simple algorithm coordinates the actions performed by the search engine: if the first "basic search" provides satisfying results, i.e. it provides a number of web resources, more relevant than the specified threshold, that is greater than the amount required, the goal is achieved and results are returned. Instead if the number of relevant resources is too low the query is iteratively expanded, concept by concept, using operators for ontology navigation, and the search is re-issued. The cycle reaches its end if a relevant set has been found or if the time required for the retrieval exceeds a given amount.

Fig 3. Shows the architecture of the proposed search engine while following subsections explain in more detail the search process.

4.2 Relevance Evaluation

The evaluation of search effectiveness by the search engine itself is a crucial step for automatic triggering of query refinement and expansion. Therefore, an effective evaluation must be available to decide if something has to be done and, possibly, what to do.

The IR model described in the previous section provides relevance values expressed as the cosine of the angle between the user query and each retrieved document. Those values are, as known, comprised between 0 and 1 with values near one meaning higher relevance (i.e. corresponding resources are close to the query vector).

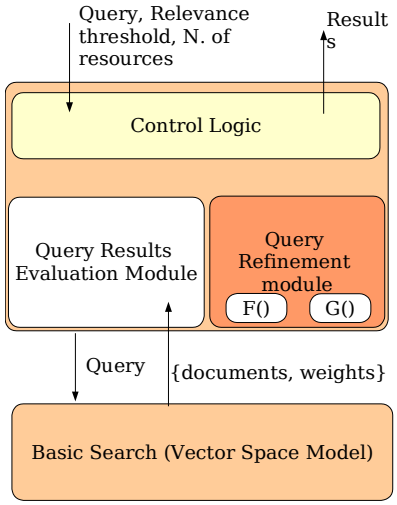

Fig. 3 Conceptual architecture for ontology based search

Starting from this simple measure a threshold based approach is set up to detect how good is the last performed search and to eventually trigger some "intelligent action" for improving retrieval performance.

A search is evaluated in terms of the number of retrieved relevant documents: if the basic search engine provides a number of good resources N_g that is equal or higher than the amount N_r specified by the external application the search outcome is considered good enough. Instead, if the number of relevant entries is lower than N_r a refining action could be started.

Def: *Let N_g be the number of relevant resources retrieved by the vector space based search engine and N_r the number of relevant resources required by the external application. The relevance evaluation heuristic is defined as:*

```
If (Ng>Nr) then
        provide results;
else
        refine the query;
```

4.3 Query refinement

Once the detection mechanism described above forces a query refinement, two actions could be done: involving the user or involving some sort of intelligent process that "simulate" user behavior.

We designed an automatic refinement tool which is able to access the underlying ontology and uses semantic relationships in order to improve search results. Its logic is strongly based on detection heuristics and ontology navigation/reasoning. Heuristics are used to define when a refinement action should be taken and what action should be performed.

Relevance evaluation is the key point of this approach since a crucial factor in this automatic process is the ability to trigger a refinement action. Whenever an action should be performed, other techniques are required to identify the kind of operation that must be done. We used again a threshold based technique because of its ability to be experimentally tuned and automatically adapted to a user profile.

Having established that an action should be performed, then the query must be analyzed, together with the result set, in order to find which concepts must be focalized, which ones must be generalized and which ones achieved a satisfying level of detail. The evaluation of relevance of concepts composing the query is as follows: for each concept the number of retrieved documents annotated with respect to that concept is evaluated. If it exceeds an internally defined ontology navigation threshold *O*, then the concept is left as it was in the original query. Instead, if the opposite situation occur, a second internal threshold *F* selects the semantic operator to be applied between focalization and generalization.

Focalization consists of moving the query from father nodes to child nodes of the ontology following *isA* relationships while generalization is a bottom-up navigation which moves the focus from leaf nodes to parents.

Def: *Let N_{rc} the number of relevant documents with respect to the concept c and O the navigation threshold. An ontology navigation action is performed if and only if*

$$N_{rc} < O \tag{5}$$

Def: *Let F be the operator selection threshold. If*

$$N_{rc} < O \wedge N_{rc} \geq F \tag{6}$$

a focalization step is performed onto the concept c, else if

$$N_{rc} < O \wedge N_{rc} < F \tag{7}$$

a generalization step is triggered.

The underlying idea is that if the number of relevant resources that have been retrieved by the concept level IR system with respect to a given concept is not

sufficient, but is still valuable, a focalization step may improve search results by reducing the search space and focalizing the query towards a more "specific" direction. On the other hand, if the number of relevant documents is negligible, the query may be generalized by capturing the available information from a more generic point of view.

The described query refinement process involves ontology navigation, leveraging semantic information for search result improvement. The mechanism is quite simple and automatically directs the search towards a good amount of relevant resources using a query expansion process: relevant concepts, extracted performing ontology navigation, are added to the original query with a weight equal to the original query concept to which they are related. This choice allows the refinement of the conceptual query without losing any original information.

5. Semantic Search Engine in DOSE

We implemented the semantic IR system described in the previous sections into the DOSE architecture [4]. DOSE is a Distributed Open Semantic Elaboration platform based on a modular multilingual architecture, which includes ontology, annotations, lexical entities and search functions. The platform is implemented as a distributed set of services including: semantic annotations for document substructures (e.g. chapters, sections, paragraphs), an external annotation repository (based on XPath and XPointer technologies) that is automatically populated starting from a known ontology and a lexical representation of concept classes, and a simple annotation search engine used to extract and recombine relevant document fragments. Annotated resources may be XML or XHTML static or dynamic documents, and need not be stored nor modified.

The architecture is implemented in Java and uses XML-RPC messages for communications among internal modules and with external applications. The proposed semantic information retrieval system, has therefore been implemented in the same language.

In order to assess the valuability of our approach we deployed the semantic IR system by defining two new DOSE modules, the first implements the *tf*/*idf* vector space model at the semantic level, while the second is used to provide "smart" functionalities for query refinement based on ontology navigation. Those two modules have been named "Basic Search" and "Clever Search" respectively and since the DOSE architecture is distributed and allows concurrent access, they cannot have memory of past queries. Therefore each refinement step requires the specification of the original query string.

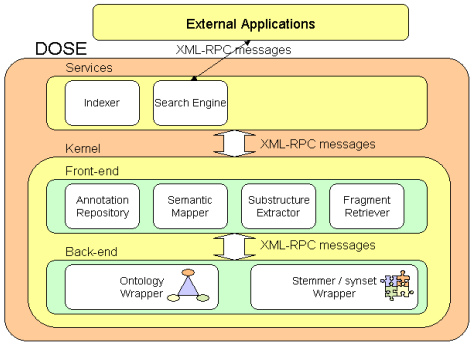

Fig. 4, DOSE architecture

The resulting architecture is depicted in Fig. 5.

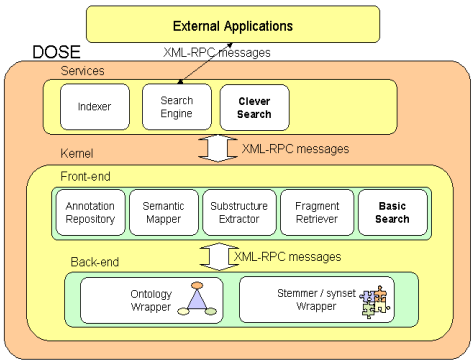

Fig. 5, Semantic Search modules in DOSE

The DOSE architecture offers a particular module called Semantic Mapper that is able to map a text fragment on to a weighted set of ontology concepts by identifying, into the fragment, lexical entities associated to ontology concepts. Lexical entities are words or word compounds that are usually adopted by humans to identify the concept to which they are attached.

The query that must be provided to the search engine could therefore be obtained from a classical query string interface and subsequently mapped into a conceptual one by using this module, otherwise the external application is charged to provide the query to the engine in the correct format.

The Basic Search module implements a *tf*/*idf* vector model on the search space composed of annotations stored into the Annotation Repository. It accepts as input a conceptual query composed by a sequence of {concept,weight} pairs, evaluates that query against the annotation set and ranks the annotations according

to a similarity value calculated with the cosine measure (1). Then it selects a number of resources equal to the one required by the external application and returns a set of {URI, Xpointer} pairs for fragment retrieval together with the global relevance value associated to each resource.

An external application which already implements a reasoning scheme can directly access the Basic Search module as a retrieval front-end for the Annotation repository, however it is likely that most applications will request a relevant retrieval relying on query refinements automatically performed by the Clever Search module.

The Clever Search module implements a query expansion mechanism based on ontology navigation. A given query is first passed to the Basic Search and then evaluated following the procedure described in 4.3. If a refinement is required the module first determines which action must be performed (6) (7) and then interacts with the ontology wrapper for effective navigation. New concepts are weighted exactly as the ones from which the refinement started and the new query is submitted to the Basic Search. The entire process can be iterated until a relevant set of resources has been found or a stop criterion has been reached.

In both cases the Clever Search provides as output the list of {URI, XPointer} pairs for relevant fragments together with corresponding relevance values. It is also possible to force the Clever Search to compose a result page by concatenating fragments into a valid XML file, in this case the module interacts with the fragment Retriever for resource fetching from the web and composes the result page. The diagram in Fig. 6 depicts a "Clever Search" scenario.

6. Experimental Results

We set up a couple of experiments in order to assess the approach valuability and to point out improvements between a common IR technique applied to semantic annotations and the ontology driven smart search engine we propose. To perform such experiments a real-world ontology has been developed in collaboration with the Passepartout service of the city of Turin.

The Passepartout service is a public service for disabled people integration and aid, and is active in social environment since 1981. The developed ontology was about disability aids, norm and laws, and social integration. It involved at least two experts from the Passepartout service and one ontology engineer. The ontology is organized on 4 different areas which are modeled in deep detail, as an example "disabled people working aids" was one of them.

At the end of the first interaction cycle the ontology counted about 450 concepts organized into 4 main areas, for each ontology concept a definition and a set of lexical entities has also been specified (see [4] for a more detailed explanation) for a total amount of over 2500 words.

The Passepartout web site has been semantically indexed using the disability ontology and the DOSE architecture; a set of 1500 web pages have been annotated at different levels of detail starting from the whole body down to the single paragraph. The resulting set of semantic annotations is composed by over 15000 annotations stored into the Annotation Repository.

Starting from this great amount of data we defined a set of queries and correspondent relevant pages. In order to make the evaluation simple but still meaningful we decided to consider as relevant all fragments coming from a document that was judged relevant.

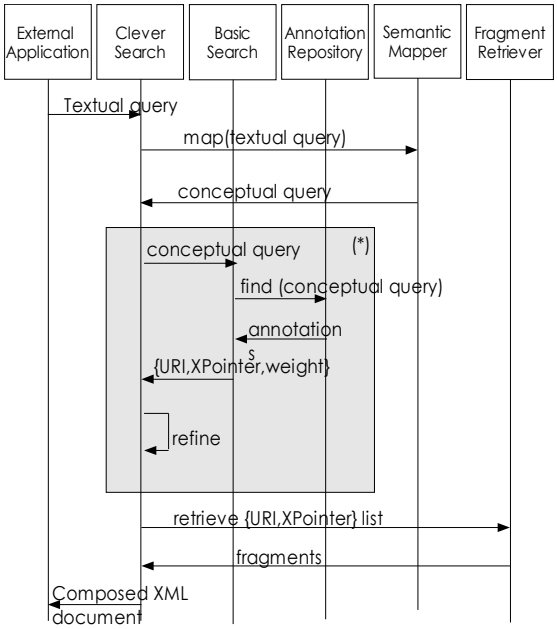

Fig. 6 Clever Search scenario for a textual query

Two different queries have been issued both to the Basic Search module and to the Clever Search and results have been compared by means of precision-recall graphs (Fig. 8 and Fig. 9).

In both cases the ontology powered search provided better results in terms of precision and recall.

Looking at the two graphs it is easy to notice that the Basic Search line ends before the Clever Search one. This is due to the fact that documents judged as relevant by a human expert were not annotated by concepts specified in to the query, in other words documents judged relevant were not annotated in the best possible way due to a lack into lexical entities definition, therefore they were non retrievable by the basic search engine. Since the Clever Search engine performs a query expansion, new concepts were introduced into the query (Fig. 7) allowing the retrieval of previously "uncoverable" documents.

Results shown in Fig. 8 and in Fig. 9 are not yet competitive from the point of view of precision and recall since the Basic Search module is not optimized, however they show that an ontology based query expansion process is able to provide improvements into search result relevance by retrieving quickly relevant documents and by discovering knowledge not explicitly expressed into the user query.

First test: original query: "Agevolazioni tariffarie" "Sordomutismo" (English): "Financial aids" "Hearing impairment" expanded query: "Agevolazioni tariffarie" "Sordomutismo" "Menomazione" (English): "Financial aids" "Hearing impairment" "Disablement"
Second test: original query: "Famiglia" "Reti informali" "Integrazione sociale" (English): "Family" "Informal networks" "Social integration" expanded query: "Famiglia" "Reti informali" "Integrazione sociale" "Reti sociali secondarie" (English): "Family" "Informal networks" "Social integration" "Second order social networks"

Fig. 7 Original queries and refinements

7. Conclusions

We proposed a semantic search engine based on query refinement powered by ontology navigation. We analyzed various techniques for query expansion at the semantic level and tested our approach on a real world scenario with a large size ontology.

Experimental results show that semantic integration of classical IR models is effective and can improve significantly retrieval performance in terms of precision and recall. In particular we demonstrated that ontology navigation can alter the ranking of retrieved resources providing quickly relevant documents to the user.

Fig. 8 First precision vs recall test

However a more extensive test must be performed and we are planning to apply our search engine to different web sites. Another interesting evaluation would be a comparison with traditional approaches on the TREC database, but this could not be done up to now due to the lack of a domain ontology for the TREC repository.

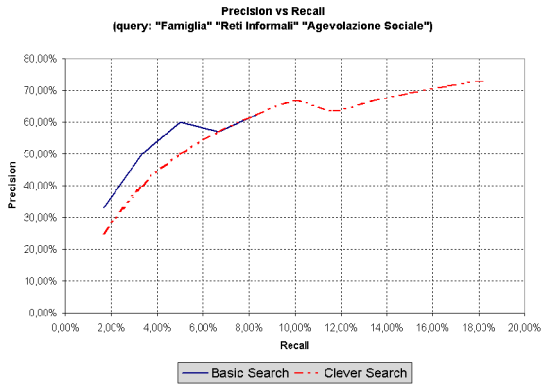

Fig. 9 Second precision vs recall test

In a near future we plan to extend the implemented modules to support the full power of semantics using ontology reasoners, also we are working on several innovative search interfaces in order to use semantics, even from the query formulation step, and in order to make resources accessible for non-expert users.

References:

[1] KAON Ontology and Semantic Web Infrastructue http://kaon.semanticweb.org

[2] R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information retrieval", Addison-Wesley, 1999.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In Ashman and Thistlewaite [2], pages 107--117. Brisbane, Australia.

[4] D. Bonino, F. Corno, F. Farinetti, "DOSE: a Distributed Open Semantic Elaboration Platform", ICTAI'03, Sacramento, California, November 2004.

[5] On-To-Knowledge-Project, http://www.onto-knowledge.org

[6] CABLE: CAse Based e-Learning for Educators, http://elite.polito.it/cable