Ontology-driven Information Retrieval in FF-Poirot

Roberto Basili, Marco Cammisa (†), Marco Pennacchiotti
Dept. of Computer Science (DISP)
University of Roma, Tor Vergata
Email: *surname*@info.uniroma2.it

Fabio Massimo Zanzotto
DISCO
University of Milano "Bicocca"
Email: zanzotto@disco.unimib.it

Dario Saracino, Maria Vittoria Marabello
(†) Knowledge Stones S.r.l.
Roma
Email: {mmarabello,dsaracino}@ais.it

Abstract— **This paper proposes a new approach for supporting domain information retrieval and information extraction on the web, using an original query expansion technique supported by an ad-hoc ontology focused on a specific domain of interest. The system has been built and tested in the framework of the FF-Poirot project, for supporting fine-grain retrieval from the Internet aiming at detecting financial fraudent sites. In a first stage, using a short list of keywords given by the user, the application mines the web retrieving relevant documents. These documents are then clustered into coherent groups focusing on specific subjects. The ontology model is devoted to represent the most important concepts of the domain of interest and to link them to the user need as expressed by the keywords. Once clusters of documents are made available after the first stage, the ontology can be used to extract from these clusters the most interesting documents (the most probable fraudolent sites in the framework of the FF-Poirot application). Browsing the ontology and selecting specific concepts, the user starts a query expansion engine that refines the search, creating a new query based on terminological evidences tied in the ontology to the selected concepts. The paper describes the overall software architecture of the application as used in the project, focusing specifically on the query exapansion engine and the supporting ontological model adopted. Experimental evidences, as emerged in FF-Poirot, will be used to prove the feasibility and the advantages of the adopted technique.**

I. Introduction

Semantic methods for Information Retrieval (IR) are inherently limited by the influence of dangerous phenomena of ambiguity and lack of coverage. Semantic Web applications are even more problematic as for the size and heterogeneity of the target data/information. In [1], a linguistically motivated ontology model that integrates domain information with a specific lexical semantic subsystem has been presented. One potential application of the model is IR as it allows to bridge flexibly the gap between ontological primitives (concepts and n-ary relations) and lexical knowledge, e.g. terminology and verb argument structures. This model has been fully exploited in the FF-Poirot project to support fine-grain retrieval from Web pages aiming to detect financial fraud. The ontology is here used to drive (and increase precision) the behaviour of meta-search engine mining the Web contents.

Specifically, positioned in the context of the Semantic Web, the FF-Poirot project (EU - IST 2001-38248, [2]) aims to build computable knowledge resources (e.g., financial and forensic anthologies) and specific methodologies and systems to support financial and legal expertise in detecting and preventing financial frauds on the Web (VAT processes, securities exchange, investments, banking and insurance services). In the framework of the project, a key role is played by technologies able to support the expert in searching and mining the web looking for potential fraudulent sites. A completely automatic and unsupervised agent for such kind of web searches is in fact too far reaching today, for two main reasons. From the one hand, knowledge and reasoning required to detect Web fraud rely on heterogeneous and complex decision making that is far beyond language processing capabilities: for example, often clues for frauds are even outside the language sphere (e.g. a not trusted web server). On the other end, such knowledge is highly dynamic, as fraudulent actors adapt their strategies to the countermoves of the financial institutions. The focus is rather on effective Web search processes that trigger the detective activities: a supporting NLP system can offer advanced linguistic and ontological capabilities to speed-up and refine the IR activity carried out by the legal and financial experts.

In this framework, the IR query expansion system proposed in FF-Poirot uses both specific ontological resources and NLP techniques in order to boost the IR activity with the linguistic knowledge needed to improve both precision and efficiency of the search. The aim is to set the search activity in an intuitive and coherent software environment (Protege [3]), in which the expert is allowed to navigate a financial ontology, and retrieve automatically documents related to one (or more) concepts that are tied to fraudulent events.

In Sec.II a short overview of the FF-Poirot application is presented, while in Sec.III an overview of the model for query expansion is given. Moreover, in Sec.IV and Sec.V the supporting ontological model and the IR query expansion engine are described in deep. A first evaluation of the engine is then presented in Sec.VI. Finally, Sec.VII outlines some conclusions and thoughts on the use of ontology-driven IR in the framework of the Semantic Web technologies and perspectives.

II. The FF-Poirot Application

The FF-Poirot application has been developed together with the CONSOB [1] expertise and is inserted in the range of tools

[1]Consob, the Italian public authority responsible for regulating the Italian securities market, plays the role of user in FF-Poirot

devoted to monitor online fraudulent activities by means of Information Retrieval systems, whose performance are empowered by the embedded use of domain-specific knowledge resources. The application is intended to support government agencies (but also services firms and corporations) in the detection and prevention of online frauds (abusive investment solicitation, unauthorized investment services execution) against investors.

The application is basically ontology-driven: the use of domain-oriented knowledge resources gives the opportunity to show how these instruments allow the implementation of powerful and yet flexible solutions, in principle portable across key applications domains in industry and trade.

A search agent has been then developed, providing instruments to perform the following tasks:

- Monitoring Web sites to look for illegal investment solicitation or unauthorized investment service offers. The selection and extraction of potentially fraudulent sites is an ontology-driven, Information Retrieval process, where the domain-specific ontology has been created by using information coming from the user context description (i.e. specific material on financial fraus domain provided by the CONSOB experts); access to ontology is used to improve the relevancy of the retrieval.
- Ontology-driven search. In the CONSOB Ontology domain concept, linguistic knowledge (word senses) and terminology are represented. The latter information (senses and terminological entries) can be used to query the mined Web material. They extend the set of keywords used to search and cluster the large amount of Web pages related to the fraud financial investments and can be interactively used by the expert both *before* and *after* the download. In the *before* modality the ontology offers the user concept-based views of the documentary knowledge, and enable the naming of the different text clusters derived according to the defined concept hierarchy and word sense network.
- Ontology-driven browsing carried out by a *query expansion engine*. After the download of documents from the Web the amount of texts to be inspected and verified is still challenging for the expert. In this phase the ontological concepts can be used to browse the mined Web material during inspection. Interactively, the expert can look at individual clusters of documents as they are made available by the IR component. Alternatively the user can just navigate individual clusters through the concept hierarchy and look for specific concepts/abstractions (e.g. *"investment bank"* or *"capital gain"*, *"net worth"*). We can call them the *target search concepts* (tsc). By selecting these tsc the user expresses his interest in focusing only on the documents dealing with those notions. The application supports the expansion of the tsc by means of all their related terms. In synthesis all the tsc and their generalizations are carefully inspected by the system and terminological entries connected with any of them are collected. This set of terms is then used by the Web search engine in a query expansion phase: documents internal to a cluster are re-ranked and form a user specific cluster. This re-ranking is helpful after the downloading to intelligently allow the system to refocus on several tsc within a single Web mining session. The ontology offers thus a sort of *semantic GUI* for Web sites inspections.

Focus of this paper is to describe the query expansion engine and the ontological model that drives the search. Indeed, such components can be seen as the basic components of a new scalable and portable solution for domain specific ontology-driven Information Retrieval, strongly based on the principles and standards of the Semantic Web. The next sections thus describes the FF-Poirot query expansion sub-system, focusing both on the ontology and the expansion engine.

III. Ontology-driven IR model

Aim of the query expansion engine, as underlined in the previous section, is to automatically refine a user query and to re-rank a cluster of documents already downloaded by the system and built by other modules of the FF-Poirot application. Each cluster represents a *Task Category* tc, and contains all documents D_{tc} related to a specific area of interest for the user (e.g. *on-line investments*). The engine thus operates on the set of task categories TC and the related set of documents D_{TC}. Task Categories are carefully integrated in an ontological model in which domain knowledge can be browsed by the user to select a specific concept, in order to re-rank documents in D_{TC} and show only those interesting document that satisfy its specific information need.

For example the expert could ask for pages related to the concept *new cooperative credit bank*, as it is a potentially fraudulent activity if present on the web. This sort of *conceptual query* is thus submitted to the system. The system should then be able to start up an IR search on D_{TC}, using the linguistic knowledge (i.e., IR keywords in form of terms related to "*new cooperative credit bank*") needed to retrieve the most promisingly fraudulent web site. At the end of the search, the retrieved sites should then be presented to the expert for a final inspection. In this framework there are two main advantages:

- The expert is not requested to build any specific query for the IR engine. Indeed, the expert must only browse the ontology looking for the desired concepts. The burden of finding the often complex linguistic and query level expressions of the information is transferred to the system.
- The linguistic knowledge encoded in the system is able to refine the query using all the linguistic material related to the specific concept, that in most cases can not be foreseen by the expert. Moreover, linguistic material can compose the final IR query in a complex weighted Boolean expressions to boost the search.

IV. The ontological model

The system architecture needs three main *knowledge layers*: an ontology (Sec.IV-B), representing generic *domain conceptual knowledge*, a corresponding *domain linguistic knowledge*, and a specific set of Task Categories (Sec.IV-A) expressing the *user* profile that drives the search.

The query expansion system (Sec.V) that builds the query for the external IR re-ranking engine works on the basis of these knowledge layers, by activating an ontology concept (*conceptual query*). The different layers are examined to extract the linguistic material to form the query: moreover, each keywords is properly weighted according to the importance it assumes in the conceptual query.

The ontology gives the main contribution in the query building process, as it contains most of the linguistic material. While its semi-automatic building process can be time consuming, it is a one time effort, as domain knowledge is quite static and fixed in time. On the other hand, task categories express a sort of user specific information need. They must then be set-up independently for each user: luckily, their building process is faster and less complex, as only little and unstructured knowledge is needed.

A. The task categories

Task categories are used to represent a user profile for the search. Each category represents a specific user information need, and simply consists in a list of keywords. Categories are built by previous stages of the application (see Sec.II): during system set-up, the user is asked to enter the list of keywords from which he would start its search. In a *conceptualization phase* keywords are then clustered semi-automatically into categories TC using lexical-semantic criteria . A knowledge engineer takes care of supervising the process. The web search engine thus browse the web retriving documents D_{tc} for each category tc. Ranking in each cluster is a side-effect of the adopted search engine. In the final CONSOB application, this initial Web mining phase triggered by about 110 user keywords (proposed by the experts).

Semantically, task categories thus represent a sort of situational areas of interest, as implicitly expressed by the user, that is, typical domain situations in which the user is interested. In order to integrate this implicit information need into the domain ontology that will support the query expansion phase, a specific *anchoring phase* is devoted to link categories to the ontology. In particular, each category is represented in the ontology by a so called *task relation* as it will be described in the next section.

For the CONSOB application 10 categories have been designed according to the user seeding information (keywords). Some examples are reported in Table I.

Task Categories	*Keywords*
FRANCHISING INVESTMENT (INVESTIMENTO IN FRANCHISING)	franchising partner iniziativa partner della iniziativa titoli titoli azionari azioni collocamento investimento
COMPANY INVESTMENT (INVESTIMENTO SOCIETARIO)	emissione obbligazionaria emissione azioni emissione quote valore nominale aumento di capitale sociale azioni privilegiate azioni ordinarie prestito obbligazionario titoli azionari
GAIN INVESTMENT (INVESTIMENTO IN QUOTE)	capitale sociale quota sociale quota associativa quota societaria dividendi azioni privilegiate azioni ordinarie prestito obbligazionario titoli azionari
ON-LINE INVESTMENT (INVESTIMENTO ON-LINE)	investimento on-line investimento on line investimento in Internet

TABLE I
EXAMPLES OF TASK CATEGORIES FOR THE CONSOB APPLICATION

B. A syntactic-semantic interface ontology

The ontology for the IR expansion system is based on the ontological model proposed in [1]. Aim of the ontology is to model a syntactic-semantic interface between a specific domain knowledge and its linguistic realizations, in the framework of the Semantic Web. The model is in fact formalized in the OWL ontology language. A bridge between the domain conceptual knowledge (called *Domain Ontology*, DCH) and their linguistic counterparts (called *Lexical Knowledge Base*, LKB) is also modelled in the ontology.

The DCH is formed by a set of domain *Concepts* and relations among them, called *Semantic Relations*. Semantic Relations define the useful (typed) relationships required by a given application. Relations usually define what is often expressed linguistically in terms of complex verb predicates. A Semantic Relation in our ontological model has a frame-like semantics. The resemblance with the notion of Frame, as used within the FrameNet project [4] is strong: indeed, Semantic Roles here corresponds to Frame Elements. In an financial application, for example, a typical Semantic Relation is *Selling* and it involves concepts like *legal entities* (companies and persons), *products*, *money* and so on. Major properties of the domain Relations are *Semantic Roles*, usually employed to characterize the concepts participating (i.e. that act as slot fillers). Semantic Roles are thus role labels for the Concepts involved in a relation. As a semantic relation r fully determines the specific concepts allowed as its fillers, legal (i.e. allowed) values for the Role slot are ontological concepts, i.e. semantic restrictions to individuals suitable as Role fillers. In this way, Selectional Restrictions are implemented as type restrictions on Role fillers. For example, a typical Semantic Role for a *Selling* relation is *Buyer*: its slot filler could be the `legal entities` concept in the DCH. Also *Good* and *Money* are roles with type restrictions as `products/shares` and `money` respectively. More formally, using a Description Logic formalism the *Selling* relation can be defined as follows:

$$\texttt{Selling} \equiv (\exists\ \texttt{hasBuyer.Legal\_Entity})$$
$$\sqcap\ (\exists\ \texttt{hasDonor.Legal\_Entity})$$
$$\sqcap\ (\exists\ \texttt{hasGood.(Share} \sqcup \texttt{Product))}$$
$$\sqcap\ (\exists\ \texttt{hasMoney.Money})$$

The DCH is then devoted (as in the traditional view on the ontology) to define properties of individuals, relations and typical task involved in the application process.

The LKB constitutes the language component including lexical semantic information: *Terms*, *Word Senses* (inspired by WordNet and making use of a consistent subset of the hyponymy hierarchy, the Wordnet Base Concepts (*WNBC*) [5] and linguistic relations (mainly *Verbal Predicates*) structured according to linguistic methods and principles and modelled independently from the domain knowledge.

DCH and LKB are mapped through specific ontological sub-hierarchy or assertions. Specifically, Concepts are mapped to Terms through a property called *related_terms*. Terms are usually unambiguous in a specific domain, so that they are mapped to a single concept; on the other hand a concept will be linguistically represented by one or more (related) terms. For example the concept *new cooperative credit bank* is mapped to its (Italians) terms through the restricted property:

$$\forall\ \texttt{related\_terms (nuova\_banca\_di\_credito\_cooperativo} \sqcup \texttt{nuovo\_banco\_agricolo\_mantovano)}$$

Semantic Relations are mapped to Verbal Predicates: as both hierarchy are formalized using the frame formalism, their mapping is more complex and requires a specific sub-hierarchy to link Semantic and Syntactic Roles.

The ontology is semi-automatically built in an incremental fashion. Starting from a minimal ontology, made available in the early phases of the ontology engineering process, an incremental process takes care of interleaving a NL learning phase (to acquire linguistic knowledge) with the ontology engineering task. The NL learning phase is carried out by linguistic systems able to automatically extract terms and verbal predicates from large corpora [6] [7]. More details can be found in [1].

Specifically, two main sub-hierarchy of the abovementioned model are used in the FF-Poirot application: Concepts and Terms. The Concepts hierarchy is used by the expert to build the conceptual query: once the needed concept is selected, all Terms linked to it are retrieved, by the *related_terms* concept's property. Terms will then be used to automatically build the actual IR query.

The original ontological model presented in [1] has been augmented with specific type of semantic relations, called *task relations*. In the FF-Poirot framework, a task relation represents a financial event/situation strictly related to the task of fraud detections that are interesting for the expert. Each relation is described by appropriate semantic roles. Semantic role values are restricted to specific ontological Concepts through *selectional restrictions* (expressed in disjunction in a *for-all* OWL restriction). Concepts used as restrictions are called *task concepts*.

All the presented layers of knowledge are finally tied together. User knowledge (task categories) is linked to domain conceptual knowledge (Concepts) through selectional restrictions. Domain conceptual knowledge is linked in turn to domain linguistic knowledge (Terms) through explicit ontological relations. The resulting semantic description enables a suitable query expansion mechanism triggered by the activation of task categories and domain concepts.

For example the task category (relation) *Investment Solicitation* describes the generic situation of financial investment solicitation carried out on the web, using specific semantic roles properly restricted to specific task concepts.

The interplay of these different layers constitute the strength and the richness of the above query expansion system.

V. The query expansion engine

Aim of the system is to create specific domain IR queries, starting from a *conceptual query* activated by the expert selecting an ontology Concept. The output should then be a list K of complex keywords (e.i., Terms) to submit to the external IR engine. Moreover, keywords should be properly weighted, to allow the IR engine to give more importance to keywords more tied to the conceptual query. Weights can range between 0 and 1.

As a driving example consider an expert looking for a fraudulent activity consisting in an investment solicitation carried out by a fake cooperative bank on the web. The expert could then activate the Concept *new cooperative credit bank*.

The algorithm, summarized in Fig.1, proceeds as follows. Given the selected Concept c, all its linked Terms T_c (*related_terms* property) are inserted in K_c, with weight 1, as they represent the keywords that better express to the conceptual query. In the example:

$$T_c = \{\texttt{nuova\_banca\_di\_credito\_cooperativo}, \texttt{nuovo\_banco\_agricolo\_mantovano}\}$$

A climb in the hierarchy then begins to examine the c ancestors. The aim is to add to K_c also terms related to the ancestors, since they can be useful to further refine the search. As more general ancestors are less significant for the original conceptual query, terms extracted from higher levels of the hierarchy receive increasingly lower weight (the weight function is described in Sec.V-A).

As the Concepts hierarchy allows for multiple-inheritance, more than one climb path can be followed. Each path stops when a *task concept* restricting a task relation tr is met. The set of task relations activated by the task concepts is called TR, while the corresponding categories is denoted by TC.

The algorithm thus forms the IR query Q as a Boolean combination of the weighted terms. Terms directly linked

```
expand_query (concept c, weight w)

terms_list = retrieve_terms(c,w)

If task_concept(c)
        task_relation tr = find_relation(c)
        term_list = term_list + retrieve_terms(tr,1)
 else
        For each ancestor(a,c)
                w = reweight(w)
                term_list = term_list + expand_query(a, w)
return term_list
```

Fig. 1. Query expansion algorithm. First call is *expand_query(c,1)*. *Reweight* function is described in Sec. V-A

to the conceptual query are the most important and receive weight 1. On the other hand, terms related to the activated task categories are also relevant, as they express the specific situation that the user had implicitly in mind activating the concept. In between there are all the terms linked to the climb-up paths. A virtuous mix of user and domain linguistic knowledge is then reached.

In the example, the concept *new cooperative credit bank* has multiple-inheritance with two parents: *cooperative credit bank* ($path_1$) and *new credit bank* ($path_2$). Terms linked to *cooperative credit bank* and *new credit bank* are added to K_c with a proper weight. Ancestors of *cooperative credit bank* in the climb path are: *bank* $\leftarrow$ *financial institution* $\leftarrow$ *financial subject*. *financial subject* is a task concept, as it restricts the roles *addresse*, *partners* and *speaker* of the task relation *investment solicitation* (ts_1). $path_1$ thus stops. On the other hand, $path_2$ continues to climb its ancestors, that are actually the same of $path_1$. No new terms are then added. All the climb paths are stopped: the overall climb thus finishes. As a result, K_c is formed by 42 weighted terms:

K_c = {nuova_banca_di_credito_cooperativo 1, nuovo_banco_agricolo_mantovano 1, banca_di_credito 0.34, nuova_banca 0.34, titolare_di_diritto 0.01, titolare_di_conto 0.01, titolare_di_azione 0.01, soggetto_pubblico 0.01, soggetto_privato 0.01, soggetto 0.01, soggetto_finanziario 0.01, istituto_finanziario 0.05, istituto_di_credito 0.05, istituto_bancario 0.05, istituto 0.05, istituzione_finanziaria 0.05, istituzione_bancaria 0.05, istituzione 0.05, investitore_istituzionale 0.05, banco_di_Sicilia 0.05, banco_ambrosiano 0.05, banca_virtuale 0.05, banca_toscana 0.05, banca_telefonica 0.05, banca_popolare 0.05, banca_nazionale 0.05, banca_locale 0.05, banca_italiana 0.05, banca_generale 0.05, banca_estera 0.05, banco_di_Roma 0.05, banco_di_rete 0.05, banco_di_marca 0.05, banco_di_gruppo 0.05, banca_depositaria 0.05, banco_con_prestiti_onLine 0.05, banca_comunitaria 0.05,banca_commerciale_italiana 0.05, banca_commerciale 0.05, banca_Antonveneta 0.05, banca_agricola_mantovana 0.05, banco 0.05,}

Finally, K_{ts_1} are added to K_c with weight 1:

K_{ts_1}={il_guadagno, i_guadagni, il_gratis, il_gratuito, il_valore_nominale, il_documento_informativo, la_registrazione_delle_azioni, il_sottoscrivere_azioni}

The resulting expanded query is processed against the set of documents D_{TC} retrieved from the Web according to the generic keywords related to the involved Task Categories ($tc \in TC$). For each tc, the set of most promising documents D_{tc} is initially retrieved from the Web by thus maximizing recall. Then, the query expansion algorithm operates on D_{TC} by selecting and re-ranking items according to the K_c set: a more precise set of documents D_Q is thus finally obtained and presented to the user.

A. Terms weighting function

During the climb of a query path, each term t_i related to a specific concept c_i is weighted according to the hierarchical distance of c_i from the concept c activated by the user. Specifically, the more generic the ancestor c_i is, the less weight its terms receive. The underlying assumption is that more general concepts are less interesting for the user, as they are less tied to the information need. The degree of generality of c_i is evaluated not only on the basis of its position in the hierarchy, but also on the number of its descendants (concept with many descendants is supposed to be more generic than a concepts with few). The weighting function is thus:

$$f(c_i) = \frac{f(c_{i-1})}{|T_i|}$$
$$f(c_0) = 1$$

where T_i is the set of terms t_i linked to c_i, $f(c_i)$ the weighting function for c_i predecessor in the climb path and c_0 is c.

VI. Use Case Evaluation

The ontology has been developed and implemented using OWL to allow a high level of interoperability in the context of the Semantic Web. Moreover, in order to effectively support the user during the *conceptual query* formation phase and the final retrieval, both the ontology and the query engine have been integrated in a common graphical interface, envisioned as a plug-in of the Protege ontology management tool. The plug-in shows the ontology to the user, which can easily browse the concept hierarchy and activate the desired conceptual query (Fig.3). The application then wakes up the query expansion engine, that processes the undelying query and starts the re-ranking process on D_{TC}. Finally, results of re-ranking are proposed to the user.

A qualitative evaluation of the query expansion engine can be drawn looking at the beneficial effect on some specific user queries. In particular, in the framework of the whole FF-Poirot application, the engine is expected to improve the accuracy in retrieving possible fraudolent sites with respect to the system without re-ranking.

As a use case we consider a tipical session in which the CONSOB expert is looking for fraudolent site on *gain investment* in which an internet user is invited to sign an agreement for acquiring quotes on a fake company. Without the use of the expansion engine, the application simply returns the cluster related to the *gain investment* Task Category tc: documents of the cluster D_{tc} are showed by the graphical interface, ranked according to a score derived from the user keywords of tc. As it can be seen in Fig.2 the linguistic knowledge embedded in the application is not specific and effective enough to retrieve interesting site. As a matter of fact, the first two ranked web pages (*www.canottierimestre.it* and *info.popcremona.it*) are not fraudolent sites.

Once the expansion engine is integrated in the application the quality of the system improves: indeed, the expert is allowed to expresses its information need in a more punctual and coherent way, by selecting the specific concept he is interested in. The expert thus selects the concept *"agreement module"*, to focus on all documents related to quotes agreement processes (Fig.3). The selection event activates the query expansion engine, which augment the linguistic knowledge for the query with the following terms:

modulo di adesione
prospetto
prospetto informativo
prospetto contabile
prospetto di quotazione
prospetto di dettaglio
prospetto riepilogativo

Moreover, the climbing algorithm terminates in the *gain investment* task categories, activating the related cluster document d_{tc}. The external IR engine thus uses the refined query to search and re-rank documents in d_{tc}. As a result, the graphical interface will show the new results (Fig.4). As expected, the best ranked documents are web sites which embed a possibly fraudolent activity. Specifically, *guardagnaresoldi.8m.com* is a web page offering dangerous offers to quotes acquisition of untrusted companies.

Several typical uses of the application, as the one described above, indicates in most cases an improvement in the retrival of fraudolent sites, revealing the beneficial effect of ontology-based methods to refine specific domain queries. Notwithstanding, an extensive evaluation, and more in general the set-up of an evaluation model, is still required in order to better verify the impact of the ontology in the general application framework.

VII. Conclusion and perspectives

This paper presented a new methodology for supporting Information Retrieval on specific domain, using a query expansion system based on an original ontological model. First experimental evidences emerged during the FF-Poirot project gave encuraging results: as a matter of facts, the accuracy of the system and the simple interface boosted and improved the process of retrieving fraudolent sites. Even if the system has to be intended as a prototipical architecture, further improvements could lead to a real and powerful Semantic Web application for mining the web and coherently organize and access the large amount of information retrieved.

Moreover, the effective use of the ontology for supporting query expansion is an interesting example of how ontology-based techniques can be succesfully exploited in the framework of IR and IE applications. Specifically, it emerges that in order to make the use on ontology effective in real applications, the represented conceptual knowledge must be strictly tied to the lexical knowledge as it emerges from domain textual material. We believe that only the integration and an explicit model of these two layers can be succesfull in bridging the gap between ontological knowledge and the world of real applications and resources. Notwithstansing, the development of automatic techniques to link conceptual and lexical knowledge are still a major challenge. As a future work we will thus focus on improving the ontology learning phase, in order to make the whole process of building the knowledge base as most automatic as possible. The use of relational knowledge, both at the conceptual and lexical level, has also to better explored. Verb and nominal relations between terms can be in fact exploited to further enrich the expansion system, as they represent the most part of the domain knowledge as enclosed in documents.

Acknowledgment

The authors would like to thank the Consob experts working to the FF-Poirot project. Their feedback and support during the whole architecture building process contributed significantly to the success of the system.

References

[1] R. Basili, M. Pennacchiotti, and F. M. Zanzotto, "Language learning and ontology engineering: an integrated model for the semantic web," in *Proceedings of 2nd International MEANING Workshop*, Trento, Italy, 2005.

[2] G. Zhao and R. Verlinden, "Ff poirot ontology development portal," in *FF Poirot Deliverable D6.1*. Brussel: STAR Lab., 2003.

[3] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu, and M. A. Musen, "Knowledge modeling at the millennium (the design and evolution of protege-2000," 1999.

[4] C. F. Baker, C. J. Fillmore, and J. B.Lowe, "The berkeley framenet project," in *In Proceedings of the COLING-ACL*, Montreal, Canada, 1998.

[5] P. Vossen, *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht: Kluwer Academic Publishers, 1998.

[6] R. Basili, M. Pazienza, and F. Zanzotto, "Acquisition of domain conceptual dictionaries via decision tree learning," in *Proceedings of the "15th European Conference on Artificial Intelligence (ECAI 2002)*, Lyon, France, 2002.

[7] R. Basili, A. Moschitti, and M. Pazienza, "Language sensitive text classification," in *In proceeding of 6th RIAO Conference (RIAO 2000), Content-Based Multimedia Information Access, Coll ge de France*, Paris, France, 2000.

Fig. 2. *Use case*. Application results without query expansion engine

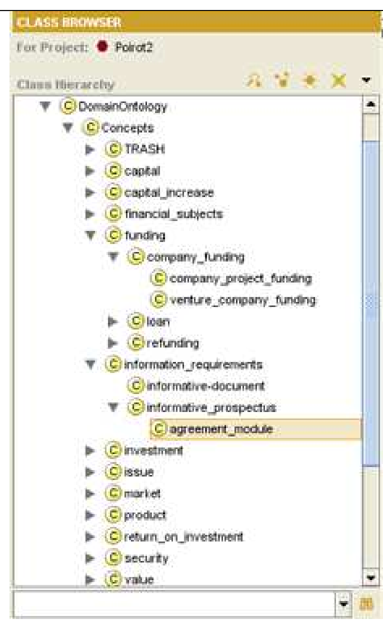

Fig. 3. *Use case*. Conceptual query

Fig. 4. *Use case*. Application results with query expansion engine