

Semantically Grounded Information Search on the WWW

Jaime BOCIO, David ISERN, Antonio MORENO, David RIAÑO

Departament d'Enginyeria Informàtica i Matemàtiques. Universitat Rovira i Virgili.

Avda. Països Catalans, 26. 43007 Tarragona, Catalunya

{jbocio,disern,amoreno,drianyo}@etse.urv.es

Abstract. The Web offers a huge amount of valuable information, but it is very hard and time consuming for a human to retrieve thousands of web pages related to a concept, filter the relevant ones, analyse their information and integrate it in the knowledge repository of a company. A Knowledge Management Platform that performs these tasks has been developed in the IST hTechSight project. This paper describes one of the components of the platform, an agent-based search module that uses the information provided by a domain ontology to find web pages that contain data which is relevant to each one of the concepts of the domain of interest.

Keywords. Ontologies, Multi-Agent Systems, Ontology-based Information Retrieval, Knowledge Management

1. Introduction

The knowledge assets of a company consist of the knowledge regarding the products, markets, technologies and organisations of a business. Knowledge adds value through a set of business processes at strategic, tactical and organisational levels. In technology intensive companies the knowledge management challenges require a tentative and cautious review of the technological domains as well as venues to monitor and assess the way in which those domains evolve, emerge, mature, and decline.

Benefits in utilising knowledge management practices include the enhancement of creativity and innovation, the strengthening of position, competence and responsiveness.

Engineers typically assess the evolution of their disciplines by reading journals, attending conferences or by hearsay. The web offers a huge amount of valuable information, but it is, weakly structured, scattered, distributed and impossible to analyse manually. Traditional search engines allow users to retrieve information by combining keywords. This type of search can cause several problems: the number of pages retrieved may not be manageable, some of the retrieved documents are irrelevant, and some of the relevant documents may have not been retrieved.

In the last years it has been argued that the performance of a search engine can be improved by using *ontologies* [1]. In its conventional form, an ontology accounts for the representation of shared concepts in a domain by specifying a hierarchy of terms facilitating communication among people (collaboration) and applications systems (integration of tools). Ontologies provide a semantic ground that can help to sort out web pages with relevant information about a concept from web pages that contain data with just syntactic similarities to the concept.

The aim of the ongoing EU research project *hTechSight* [2] is the construction of a knowledge management platform (KMP) [3] that may be used by knowledge-intensive

industries to keep a dynamically updated *knowledge map* of their domain of expertise. This paper describes in detail one of the main components of the KMP, the *search module*, whose main task is to find the web pages that contain relevant information about the basic concepts of a predefined domain represented by a *domain ontology* [4]. Several methods and techniques were developed to allow the use of the information provided by a domain ontology in order to evolve from a purely syntactic keyword-based web search as the one performed by Google or Altavista to a *semantically grounded search* as the one performed by our module. Furthermore, the search process that our system implements can be *distributed* along a computer network, in order to improve the efficiency of this time consuming task.

The rest of the paper is organised as it follows. In the next section we describe the elements that define the search process and the architecture of the agent-based search module, with a brief explanation of the different types of agents and the interactions between them. In section 3 we provide some examples of the use of the system and comment the results obtained. The paper finishes with some conclusions and acknowledgements.

2. The search module

The aim of the search module is to find the web pages which are relevant to a given domain of interest. The user of the system must provide as input a *domain ontology* as well as a set of parameters that define the search procedure. The ontology contains the concepts of the search domain and the main features of each one of those concepts, and the parameters allow the user to adapt the search to his particular needs.

The implementation of the search module is based on the *agent technology* [5] that warrants a kind of search where several software agents work together in an asynchronous, concurrent and intelligent way that can be distributed whether several computers are available.

2.1. Defining the search: domain ontology and search parameters

The concepts and features of the search domain are captured in a domain ontology that represents the first ones as classes in a hierarchical class-subclass structure, and the second ones as slots of the classes. A class and all its ancestors define a *class path*. Each class contains its own slots and it inherits all those which are defined in the ancestors [1]. For instance, fig.1 depicts a domain ontology about a subset of machine learning technologies where the classes are labelled as C, and the slots as S. So, the term "machine learning" is represented by a class with slots "dataset", "domain OR field", and "resources". During the search process, the difference between classes and properties is that the classes define the search domain, and the slots the point of view of the domain that the user is interested in. Therefore, the names of the classes are used to find the web pages which are related to the search domain, and the names of the slots in a class are used to evaluate to what extent the retrieved pages have the kind of information that the user wants to obtain. For instance, the Machine Learning ontology in fig. 1 is defined to indicate that the user is interested in the "machine learning" domain, but preferably on those particular pages concerning "dataset", "domain" and "resources", and not on pages about other subjects as "software", "people", "applications", etc.

The use of synonyms in the definition of classes and slots extends the domain ontology with the possibilities of having alternative terms as, for example, "domain" and "field", acronyms as "case-based reasoning" and "CBR", or other slang or language differences as "generalisation" and "generalization".

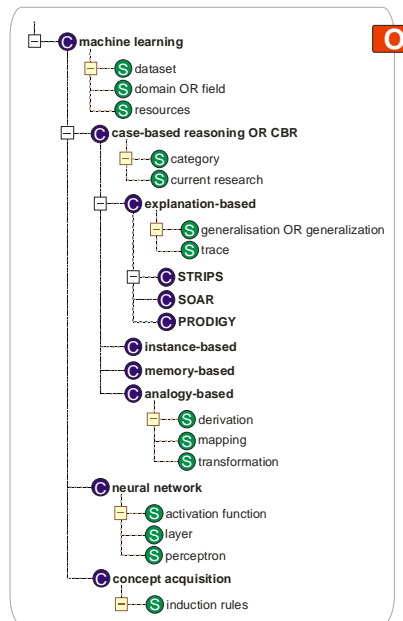


Figure 1. Machine Learning ontology as a search domain

Before the searching process starts, the user is able to adjust some parameters that affect the whole search. Table 1 supplies a brief description of these parameters, and shows their default values.

Table 1. Parameters considered in the search

Parameter	Default	Description
Deadline	500	Maximum number of seconds that the system is able to use during the search.
SearchEngine	google	Search engine to be used. Currently, the system may use either Google or Altavista.
MaxLinks	50	Number of links to be retrieved by the search engine, for each concept of the ontology.
Threshold	0	Minimum relevance value that a page must attain in order to appear in the results.
Depth	0	Search depth level.
Web site	-	Constraint on the site where the search starts in.
Language	-	Constraint on the language of the retrieved pages.
Country	-	Constraint on the country of the retrieved pages.
Internet Agents	5	Level of concurrency for the search process.

2.2. The search process

Once the domain ontology and the parameters have been defined by the user, the asynchronous, concurrent and intelligent search process can start. This is a complex process that combines several stages: splitting the domain ontology, retrieving the web pages, rating and filtering the retrieved pages, and joining the results.

During the *splitting* stage, each class of the domain ontology defines a smaller ontology which contains not only the class itself but also its class path; this sort of ontology is called *query ontology*. Each one of the classes obtained after the decomposition of the ontology is employed by an asynchronous concurrent search process that uses the names of the above classes as the keywords that define a *query*.

For each one of these queries, a set of web pages is retrieved using the search engine indicated in the *SearchEngine* parameter. The constraints *Web site*, *Language* and *Country* are also taken into account in the search. If the number of web pages retrieved does not reach the value contained in the *MaxLinks* parameter, the system raises a complementary process to increase the number of pages. This process is based in a weighted expansion tree that is built up from the initial query, as fig. 2 depicts for the class *STRIPS*. The building process is as it follows, each node of the tree is expanded with sub-nodes representing queries where one of the keywords in the parent query has been removed, except the

keyword that represent the name of the current class. For instance, the right bottom side of fig. 2 shows the list of the keywords that are in the query related to the class STRIPS. Observe that only the initial letters of the keywords are displayed in the figure. When one of the antecedents of STRIPS (i.e. "machine learning", "case-based reasoning", or "explanation-based") is removed from the initial query, the nodes A, B, C are respectively expanded. The figure also shows how the keyword "STRIPS", represented by the letter S, is in all three sub-nodes. Finally, the numbers in the nodes indicate the amount of web pages that the search system can find using all the keywords in the node.

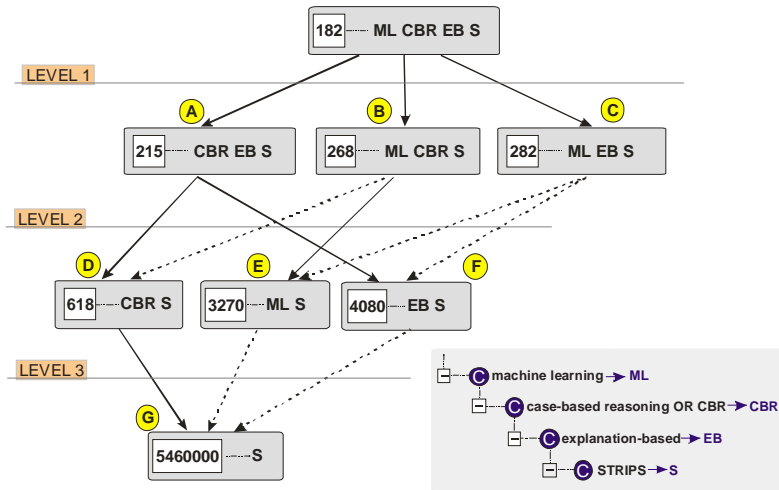


Figure 2. Best First search developed by the Weight Agent

An alternative way to extend the number of web pages to reach the *MaxLinks* parameter is to use the *Depth* parameter. This value indicates to what extend the search process takes into consideration the links contained in the retrieved pages. So, if *Depth* is 0 only the pages recovered by the search engine are considered, but if *Depth* is set to 1 the recovered pages are not only the ones obtained with depth zero, but also those other pages that are directly linked from those ones. *Depth* values above two are not recommended because the time required increases exponentially and it is not proven that the results improve.

The third stage of the search process is *rating and filtering* the retrieved pages. All the pages obtained after the retrieving process are rated according to the relevance of the query ontology to which they are related. The rate is calculated with the next function:

$$R_c(p, A) = \frac{\text{number of attributes encountered}(p, A) \times 100}{\text{total number of attributes}(A)}$$

If p stands for the web page recovered for a class C whose rate is been calculated, and A is the set of attributes (inherited or not) of C , the *attributes encountered* are the ones in A that appear in the page p . $R_c(p, A)$ defines the relevance of the web page p with respect to the class C and, after normalising it in the range $[0,1]$, it is used to rank the retrieved pages and to discard those which are below the value contained in the *Threshold* parameter, during the filtering step.

At the end, all the pages obtained for all the classes in the domain ontology are *joined* in a single structure that contains each single page as an instance of the class in the ontology. More concretely, only the URL and some other interesting information of the web pages as the title, author, keywords, date of last modification, etc. are stored in the final domain ontology.

2.3. The architecture of the system

The previously mentioned search process has been implemented as a multi-agent system (MAS) using the JADE environment [6] and following the architecture that fig. 1 shows. Four kinds of agent were identified: a *User Agent* (UA) which interacts with the user of the search system, some *Internet Agents* (IA) which access the web and analyse the web pages during the retrieving, rating and filtering stages, a *Weight Agent* (WA) that supports the search and supplies alternative queries when an IA do not reach the total amount of pages that the UA says, and a *Co-ordinator Agent* (CA) that co-ordinates the search process during the stages of splitting and joining.

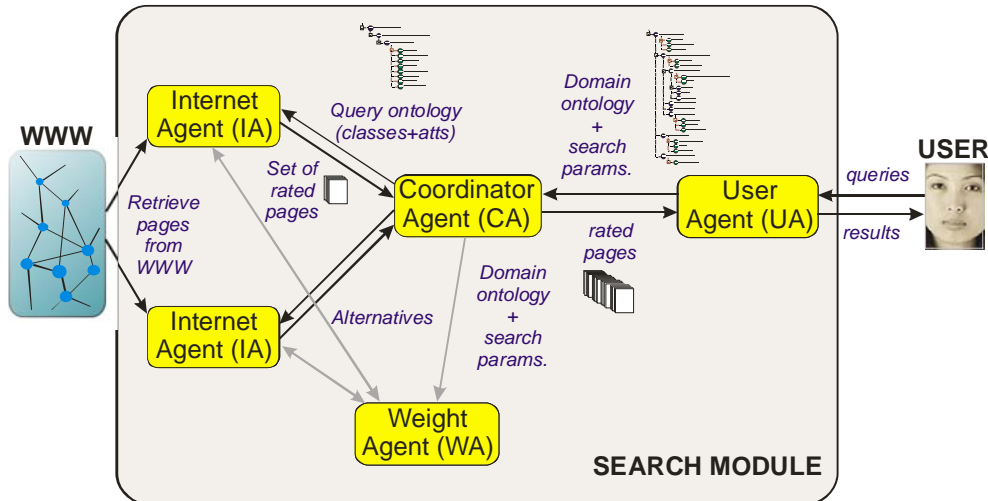


Figure 3. Multi-Agent system architecture

The way these agents collaborate in the search process can be explained following the interactions contained in fig. 3:

- The user interacts with the system through the UA, providing the *domain ontology* and setting up the *parameters* that were described in section 2.
- Once the user starts the search process by means of the UA, the CA receives and divides the domain ontology into the *query ontologies* that are spread out across all the available IAs (see the parameter *Internet Agents*).
- Each IA uses the standard keyword-based search engine indicated in the *SearchEngine* parameter to retrieve web pages that are related to the *query ontologies* received from the CA. The agent uses the semantic information of the ontology to filter and to sort these pages, and sends them back to the CA. If the IA do not recover enough pages, the WA is asked to supply alternative queries to complete the list of web pages.
- In this case, the WA explores the weighted expansion tree related to the target class, and proposes the most restrictive query, i.e. the one related to the node with less pages recovered. For instance, in the expansion tree of fig. 2, the alternative query at level two would be the one in the node A because it returns 215 web pages which is below the 268 and 282 pages recovered by the queries in the nodes B and C.
- Finally, the CA waits for all the IAs to supply the results or until the value in the *Deadline* parameter is met. Then it joins all the results and send the final unified ontology to the UA, who shows it to the user.

Both the input and the output of the search process are handled using a web interface where the user can edit, create, retrieve, import and export domain ontologies as Resource Description Format (RDF) files [7, 8]. The CA can maintain conversations with different UAs at the same time while several IAs can be working simultaneously. This way of working defines a search module which is asynchronous, concurrent, intelligent and optionally distributed.

3. Tests and Results

Two sorts of tests are supplied in this section. On the one hand, the search module has been tested to study the evolution of the relevance of the retrieved pages as some of the input parameters vary. On the other hand, an ontology in the field of *Machine Learning* has been defined and used as input domain ontology of the search module in order to conclude which are the most important web pages of this domain for some particular values of the input parameters *Web site*, *Language* and *Country*.

3.1. The analysis of the relevance

The system has been extensively tested on different domains such as chemical engineering [4]. We can conclude that the quality of the retrieved pages depends on many factors such as the appropriateness of the terms in the ontology classes and attributes, the presence of the domain in the web, the number of attributes of the class, and the input parameters that were described previously.

Three main outcomes arise from the analysis of the results. Firstly, the lack of attributes in a class produces a deceptive appearance of high relevance in comparison to other classes with attributes. A deeper analysis concludes that the reason for this fact is that the search is less demanding as the number of attributes decreases. Secondly, the average quality of the pages shows a slight descending slope as the number of pages required is increased. This confirms that as the MAS is forced to recover more pages; it has to admit new pages which are progressively less relevant than the previous ones. Finally, it is also interesting to observe that the average relevance of the pages related to a class uses to be higher than the average relevance of the pages of its subclasses, but the difference is smaller as we move down the ontology hierarchy. Like in the first outcome the reason is that, on the one hand as the classes become more specific they are more restrictive and less good pages are found. On the other hand, the number of recovered pages is kept constant for all the classes, those which are more restrictive have to admit less relevant pages, and the average rate descends.

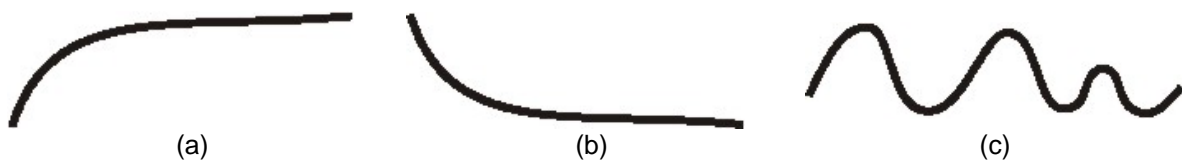


Figure 4. Models of the concept rate average

When the relevance is normalised within each class to the range [0,1] dividing it by the highest relevance obtained by a page of a particular class, we can empirically observe that the tendency of the normalised relevance follows one of the three models depicted in fig. 4. These functions represent general, specific, and WA-aided concepts, respectively:

- *General concepts* are concepts described with terms that use to appear in the web pages. Among these pages, we may distinguish between those which are positively containing information related to the concept, and those other pages which contain similar terms but related to a different concept (i.e. homonyms). Fig.4(a) shows the

evolution of the normalised average rate of general concepts. The interpretation is that as the amount of pages retrieved increases, the normalised average rate increases asymptotically to a value that represents the proportion of true positives among all the pages containing the terms used to describe the general concept.

- *Specific concepts* are concepts that either they are not very available in the web or they are described by means of precise and unfrequent terms. As fig.4(b) shows, in both cases, the first pages retrieved have a normalised average rate that decreases as more pages are required. The reason is that the few available pages about the concept are retrieved at the beginning, but if more pages are reclaimed the system is forced to retrieve less relevant pages.
- *WA-aided concepts* describe concepts that are intrinsically difficult to represent and to search because they are the result of different combinations of terms in a set. Whenever the WA proposes a new search, the first retrieved pages will have a high normalised relevance that increases to a maximum average value when the best pages have been already retrieved. If more pages are asked, the mean relevance starts decreasing until the WA is required to supply a new alternative search increasing the average rate once again. This oscillation of the normalised average relevance is represented by lines as the one depicted in fig.4(c).

3.2. The analysis of the Machine Learning ontology

A subset of technologies, tools and languages for Machine Learning has been taken into account to define an ontology. This ontology contains twelve classes and fourteen slots, as fig. 1 depicts. Some synonyms have been also defined either for classes and slots. This ontology has been used to test the input parameters *WebSite* and *Country*. For the first parameter, we have run the search in different organisations: Institut d'Investigació en Intel·ligència Artificial (IIIA), Universitat Politècnica de Catalunya (UPC), Universitat Rovira i Virgili (URV), Universitat de Lleida (UdL) i Universitat de Girona (UdG).

For the *IIIA*, the system identified the ECML2000 and Agents99 conferences as related to ML, some personal pages were also retrieved as closely related to some particular ML fields. For example, Dr. Arcos, Dr. Armengol and Mr. Ontañón are somehow related to CBR; Dr. Plaza to memory-based, SOAR, analogy-based ML, and IBL; Dr. Mantaras and Dr. Torra slightly related to NN, etc. Some research projects were also detected related to CBR (COMRIS, AMP), and many papers concerning many of the classes in the ontology. Nothing was found for the class concept acquisition. Additionally, the system indicated that some of the above relationships had a low relevance, meaning that some of the concepts in the ontology are not well covered in the *IIIA* web pages.

In the *UPC* domain, the concept ML was found in two groups of pages, some related to the research group KEMLG in general, and some others related to some particular activities of some people as the web page on ML & CBR sites and resources maintained by Dr. Sànchez-Marré or the teaching of Dr. Márquez. All of them with low relevance because they do not contain the slots in the ML class. The system was unable to detect relevant pages on NN or CBR at the UPC. Some other concepts as explanation-based ML, SOAR, PRODIGY, STRIPS and analogy-based ML were only found in web pages related to tutorials, PhD courses, technical reports and papers.

At the *URV* pages related to the research group on artificial intelligence and personal pages of Dr. Riaño and Dr. Valls were found for the concept ML. Another research group was also found related to the NN concept, and a publication of Dr. Serratosa was about CBR with a high relevance rate.

For the *UdL* the system was unable to find relevant web pages in the domain of ML.

For the UdG some personal pages close to Machine Learning such as Dr. López and Dr. Roda were found. Also, Dr. Arteaga is close to NN. Moreover, Dr. Martí, Dr. Montané and some related material about seminars to CBR were analysed.

For the *Country* parameter we have tested Spain. The most of the results are related to universities and research groups from Spain. If we use the ontology without any of the above constraints, the system also found pages related to research groups and universities, including some Spanish groups as IIIA.

4. Conclusions and Acknowledgements

Broadly speaking, we can conclude that the system is able to detect web pages related to a search domain ontology and calculate the relevance of each one of the retrieved pages. A multiagent system has been developed to make the whole process asynchronous, concurrent, intelligent and distributed. Additionally, the system has some parameters that permit the user to control some aspects of the search. The search system has been tested to analyse the variations on the relevance of the pages as the user parameters *Deadline*, *SearchEngine*, *MaxLinks*, *Threshold* and *Depth* are modified. The system has been also tested to analyse the web pages in several organisations (universities and a research institute) according to a Machine Learning ontology. The results show that the system is able to retrieve with different degree of relevance pages about related conferences, papers and other documents, PhD courses, research groups, and home pages of researches. Once analysed the pages retrieved, it is possible to have a clear idea of the people and groups working in a particular area of ML and the quality of the work they do, according to the relevance of the pages retrieved.

This work has been funded by the EU Project hTechSight [2]. We want to acknowledge the feedback and suggestions of David Sánchez.

References

- [1] D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Heidelberg, 2001.
- [2] hTechSight, "IST Project (IST-2001-33174)," 2001.
- [3] M. Stollberg, A. Zhdanova, D. Fensel, "H-TechSight: a Next Generation Knowledge Management Platform," *Journal of Information and Knowledge Management*, 3 (1): 47-66, 2004.
- [4] A. Aldea, R. Bañares-Alcántara, J. Bocio, J. Gramajo, D. Isern, L. Jiménez, A. Kokossis, A. Moreno, D. Riaño, "An Ontology-based Knowledge Management Platform," *Workshop IIWEB'03 at IJCAI'03*, Acapulco, Mexico: 177-182, 2003.
- [5] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley and Sons Ltd, 2002.
- [6] F. Bellifemine, A. Poggi, G. Rimassa, "Developing Multi-Agent Systems with a FIPA compliant framework," *Software Practice and Experience*, 31: 103-128, 2001.
- [7] W3C, "Resource Description Framework (RDF)," 2001.
- [8] A. Gómez-Pérez, M. Fernández-López, O. Corcho, *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and Semantic Web (Advanced Information and Knowledge Processing)*, Springer Verlag, 2004.