Semantic Search in the WWW Supported by a Cognitive Model

Katia Wechsler[1], Jorge Baier[1], Miguel Nussbaum[1], and Ricardo Baeza-Yates[2]

[1] Departamento de Ciencia de la Computacion,
Pontificia Universidad Católica de Chile,
Casilla 306, Santiago, Chile
kwechsle@puc.cl, {jabaier,mn}@ing.puc.cl
[2] Departamento de Ciencias de la Computacion,
Universidad de Chile,
Blanco Encalada 2120, Santiago, Chile
rbaeza@dcc.uchile.cl

Abstract. Most users of the WWW want their searches to be effective. Currently, there exists a wide variety of efficient syntactic tools that have can be used for search in the WWW. With the continuous increase in the amount of information, effective search will not be possible in the future only with syntactic tools. On the other hand, people have remarkable abilities at the moment of retrieving and acquiring information. For example, a librarian is capable of knowing, with great precision, what a client seeks by asking a small set of questions. Motivated by the efficiency of that process, we have created a web search system prototype based on ontologies that uses a cognitive model of the process of human information acquisition. We have built a prototype of a search system whose output better meets the expectations of the users compared to tools based only on syntax. Using this model, the prototype "understands" better what the user is looking for.

Keywords: Semantic Search in the WWW, Cognitive Models, Semantic Web.

1 Introduction

The explosion in the amount of information in the WWW occurred in the last years is one of the central problems faced by its users. The amount of information is so big that it is necessary to possess efficient and effective mechanisms of information recovery.

Currently, one of the most commonly used tool for search is information filtering [1], which consists of filtering the information returned by the search process based on certain parameters. Among these parameters are the user's profile [2], which is a definition of characteristics that represent the interests of the user that enables to delimit his search, such as likes, language, and interests. Another parameter frequently used is ranking, that is a score given to the web pages and depends on how many times the page has been chosen by users of the search engine. Another tool used for search is indexation [3,4], that consists of elimination of frequently used terms and search of roots of words or synonymous, to create an index that represents a document.

Q. Li, G. Wang, and L. Feng (Eds.): WAIM 2004, LNCS 3129, pp. 315–324, 2004.

All the methods described above use syntactic handling of words, in the sense that all information used from words depends exclusively on the word itself. Nevertheless, this is not sufficient; most users get frustrated when the result of their search includes plenty of pages that have nothing to do with their interests. Moreover, the amount of available information increases and systems based on syntactic methods will soon be surpassed. It is necessary, then, to add semantic elements to search systems. In this work we are concerned on how to incorporate the sense that users give to words into a search algorithm.

The absence of semantic capabilities imply that search engines return irrelevant information with words with the same syntax, but that do not mean the same thing, words that mean the same thing but appear in a context different to the user's, and leaving out those that mean the same thing but are spelled differently and appear in the same context.

An example of a semantic search system is OBSERVER [5], which uses specific ontologies that enable users to express the demanded information at an abstraction level beyond words itself. To this end, it keeps a set of small ontologies, each of them associated to a set of documents. When a requirement is demanded, the system chooses the ontology class which meets the user's requirements and asks the user for confirmation, then it returns the documents associated to that class.

On the other hand, it is remarkable the ability of the human being when look for information. For example, a librarian is capable of knowing, with great precision, what a client looks for by asking a very small set of questions. This is due, among other things, to the fact that humans easily understand the language, context and motivation of the person who asks for his help. The cognitive process of information has been studied by diverse authors [6], nevertheless, according to our knowledge, there have not been attempts of integrating cognitive models into automated search systems.

In the following sections, we describe a search prototype based on ontologies that uses a cognitive model of the human process of information acquisition. Section 2 presents a description of tools used in the prototype development, section 3 describes a cognitive model for human understanding, section 4 explains the prototype and shows a real example using ontologies of public domain and, finally, in section 5, we comment on some preliminary results and sketch our future work.

2 Ontologies, Web Documents, and Information Extraction

The following subsections describe ontologies and how they are related to web documents. Moreover, we describe the process of information extraction, which is central to our algorithms.

2.1 Ontologies and Web Documents

Ontologies provide a way to represent and share knowledge using a common vocabulary; they define a protocol of communication and allow knowledge reuse. Ontologies are useful to represent both concepts and the relationships among them.

The elements that compose an ontology are *classes*, which are a formalization of concepts; *relations* that represent interactions among classes *functions*; *Instances*, that are objects of a class; and *axioms*, that are declared logical truths that hold about elements of an ontology.

A principle of well-designed ontologies is that similar concepts be grouped together or represented using same primitive [7]. This principle is central to this work, since the closer the concepts in the graph, the more related they are.

Ontologies are not related to the web in a natural way; therefore, additional efforts are necessary to relate them. There exist two ways of relating them: the first one, is that every web page contains a small ontology that represents its content (this ontology can be written in any of RDF, OIL or DAML+OIL languages [8]). This approach gives too much freedom to the user to describe his pages, therefore making it difficult for a computer system to analyze its content. Furthermore, common users of web pages are generally unenthusiastic to use tools that define semantically well their pages.

The second way of relating web pages to ontologies, is to have a unique ontology in which every class is linked to a database of web information (web pages, pdf, images, etc.). This is achieved by creating databases containing tuples of documents associated with the classes of the ontology. In this work, we use this approach.

Currently, the CYC project [9], is a research project carrying out construction of ontologies. The CYC project is still in development, and its goal is to construct a knowledgebase containing a significant part of commonsense knowledge of this century.

2.2 Information Extraction of a Document in Natural Language

Whilst information recovery is used to retrieve documents that contain particular words, an information extraction (IE) algorithm allows a user to obtain all the concepts that could be associated to the words in the text [10].

There exist several tools that allow IE; among the GATE software [11]. These systems can work also with compound terms. Thus, such a system would be able to recognize the compound term “President of Chile” as one concept. In case a word or compound term is associated to more than one concept, all the associated concepts are returned.

In our prototype we use a Java library for IE known as Stand-Alone Gazetteer [12]. We have adapted Stand-Alone Gazetteer to work with the CYC ontology.

3 A Cognitive Model for Human Understanding

The cognitive model of information extraction (fig. 1) [13,14,15,16] consists of 5 parts: perception, observation, interpretation, understanding and reflection.

The first part of the process addresses the way information is acquired from reality, i.e. environment of the human being. We define *perception* as the act of capturing the information in the reality. Human beings have a finite capacity of storing and gathering perceived information. Humans only perceive information of their interest. That

information is determined by what is relevant for the person. For example, someone who has interest in plants, after entering to an office, would perceive the existence of them. On the other hand, one that does not have any interest in plants, would not notice them.

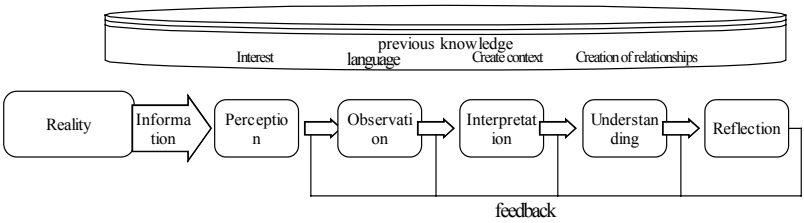

Fig. 1. Stages of the model

Perception is the act of describing an object with a language. Not all the things that are perceived are observed. For the observation of an object to occur there is a need for a language that allows to describe it. Two persons who dominate different languages will observe different things while looking at the same object. In this way, while looking to a plant, a gardener sees more things than a person who does not dominate a language that describes the plants in detail.

The interpretation occurs when the observed object is contextualized by the observer. Using the description of the object, the observer places it in a context and is able to relate it to other elements that belong to the same context. For example, if the gardener sees a plant he may know what type of plant is, if it is eatable, ornamental, etc. On the other hand, if an interior decorator sees the plant, he will be able to know where to place it, for what types of decorations it is suitable to, etc.

Understanding arises once information has been interpreted. It is a process that is strongly linked to what the observer *can do* with his interpretation in order to validate or invalidate his observation. For example, once a plant is observed, the gardener will know if it lacks light or water and will do what is necessary. The act of understanding is related with doing. Often, when a person understands something, it acts according to this understanding.

Reflection is the last part of the model. It consists of checking the whole process and determining if the actions of the observer were adequate. For example, the gardener will be able to see if the faded plant has recovered after watering it. Otherwise, he understands that it was not lacking water but another element. Thus, he increases the possibilities of succeeding in the future by repairing the information of the mistaken component.

4 Search System Prototype Based on the Cognitive Model

This section presents a search system prototype based on the cognitive model presented in the section 3. The prototype is based on the ontology cyc.daml of the CYC project.

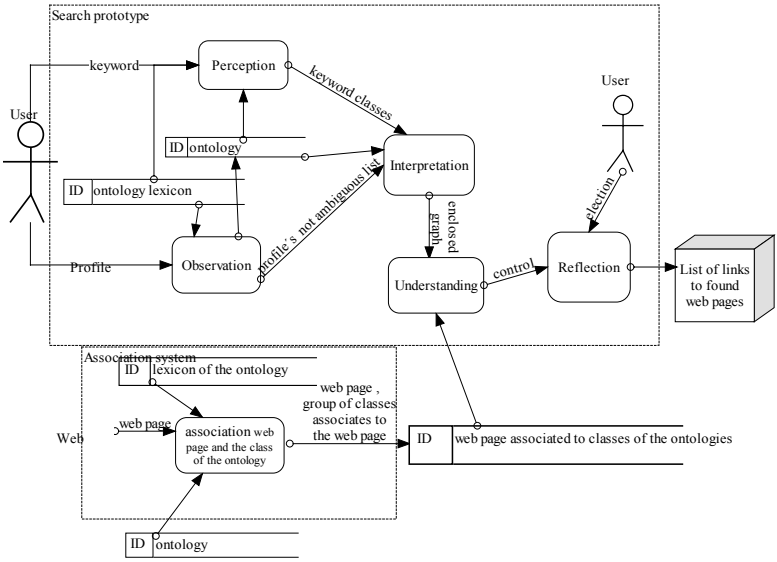

Fig. 2. Design of the developed prototype

Figure 2 shows the search system design based on the model of section 3. It is composed by five modules that interact with the user, the ontology, or the ontology lexicon, and give as result the feedback of the system and the web pages related to the search.

Besides from the system of search, there is another subsystem: the association system. Its goal is to associate the web pages with the classes of the ontology, keeping a database of pages associated to ontology classes. It must be working all the time, constantly updating its associations.

The following subsections explain and exemplify the algorithms of each of the modules.

4.1 Step 1: Perception Level

The aim of this step (first step of figure 2) is to determine what is the motivation of the system, i.e., what the system "wants" to look for. The algorithm is shown is figure 3.

```
Asks the user for a list of words
Keyword ← classes of the ontology associated those words
```

Fig. 3. Algorithm for perception level

For example, let us suppose that the system receives the word "shape". This word is associated with four classes of the CYC ontology: GeometricallyDescribableThing, ShapeType, ShapingSomething, shapeOfObject, which are stored in the array *Keyword*. This array is the output of the module.

4.2 Step 2: Observation Level

The observation level reproduces the human ability to describe a perception by means of language. In our system, it is necessary to know the language[1] of the user to understand what information is he really interested in. The system knows the language of the user by asking him to describe his interests using a paragraph in natural language.. Then, the system turns the paragraph into a list of references to classes of the ontology that represents the interests of the user.

```
List ← {<class,word> | word is in Paragraph given by user and class
                       is returned by Gazetteer[word]}
Profile_repeated ← <class, word> elements of List such that word has
                       more than one associated class in List.
Profile ← elements <class, word> of List such that word has a unique
                       associated class in List.
for each element <class, word> in Profile_repeated
    Evaluation[class, word] ← classes(nodes) of the smallest
          subgraph of the ontology generated using a BFS search
          starting from class and that contains an element in
          Profile.
endfor
for each element <class, word> in Profile_repeated
    relation_index[class, word] <- relation index of class with re-
                       gard to Evaluation[class, word]
 endfor
for each word wd such that there exists <class,wd> in Pro-
                       file_repeated
    Add to Profile Argmim_class relation_index[class, wd]
endfor
```

Fig. 4. Algorithm for observation level

When the Gazetteer recognizes a word, there exist 2 possibilities: that the word is associated with exactly one class of the ontology or that it is associated with more than one class. In spite of having more than one ontological meaning, the latter type of words have only one meaning for the user, which should be determined by the content of the paragraph (the context given by the paragraph). Due to this, for each of these words, the algorithm must determine a unique class of the ontology associated to it, considering the content of the whole paragraph.

To this end, the algorithm separates the words associated with only one class of the ontology and it places these classes in the vector *Profile*. Words associated with more than one class, are placed in *Profile_repeated*, paired to every class to which they are related.

To determine the unique class associated with every word in *Profile_repeated*, we choose the class that is semantically closer and more related to the elements of *Profile*. To measure how close and related is a particular class with the elements of Profile, the algorithm uses an *relation index*. To calculate this index, it takes into account two values. The first one, is the minimum distance between the class and some element of *Profile* (this distance can be calculated directly from the information in the

[1] The word language here must be understood in the same way as in section 3, i.e. as the means the person has to describe his knowledge.

matrix *Evaluation*). The second one, is the number of classes of *Profile* that are related to the class at minimum distance. Finally, the relation index is computed as the quotient between these two quantities.

For every word that is associated with more than one class of the ontology, the algorithm chooses the class with the minimum relation index and adds it to *Profile*. In this way, we obtain a list of non ambiguous classes in *Profile* for a given paragraph.

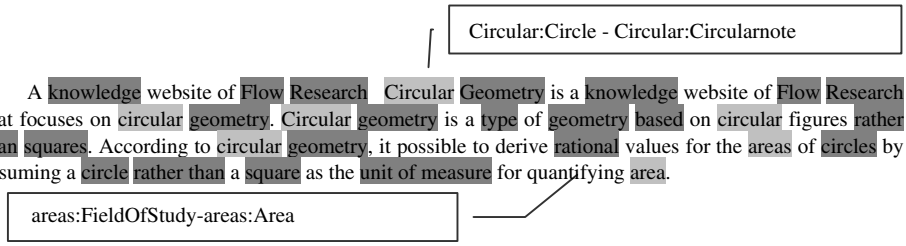

Fig. 5. Paragraph of interests that shows the words that are present in the lexicon

For example, suppose the user enters the paragraph shown in figure 5. Words in light gray have more than one associated class in the ontology CYC, whereas the words in dark gray only have associated class (words not marked are not known by the ontology).

Figure 6 shows how related are the concepts associated to the words *circular* and *area* with the rest of the words in the paragraph. In the graph, it is clear that *area* and *circle* are the classes that really correspond to the meaning given in the paragraph, since the relation index is the minimum (the relation index of circularnote is 1, whereas that of circle is ½). Thus, we leave in the profile the concepts: Circle, Area, discarding FieldOfStudy and Circularnote.

For this algorithm to work correctly, it is essential that there exist words associated with only one class of the ontology in the paragraph provided. Although this is an important limitation, it is difficult to find paragraphs that describe areas of interest and that are completely ambiguous. In case that this situation arises, the algorithm outputs an empty profile

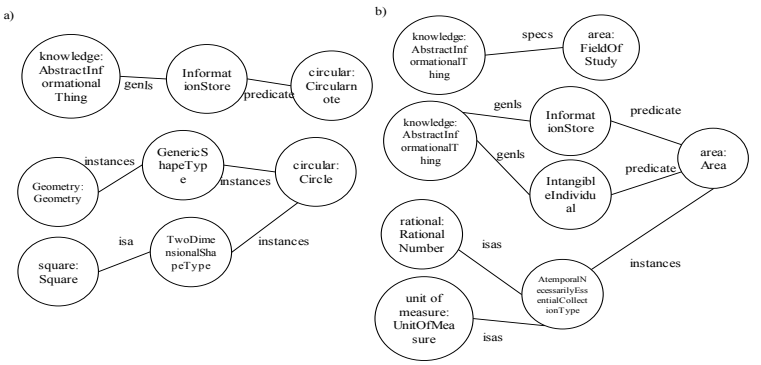

Fig. 6. (a) Search of the class most related to the word circular (b) Search of the class most related to the word area

4.3 Associating Web Pages to Classes of Ontology

For the search system to work, it is necessary to have a database of web documents associated with concepts in the ontology (shown in figure 2). To construct the database, the module uses the algorithm described in the step above. Thus, it receives a document as input (which is obtained by means of a web spider) and one association between the webpage and each class found in the document is stored in the database.

4.4 Step 3: Interpretation

In this step (interpretation module in figure 2) the keyword is contextualized, i.e., the algorithm finds the context of the word with respect to the user. Specifically, the algorithm finds the ontology subgraph that is semantically closer to both the keyword and the elements in the profile. This resulting subgraph corresponds to the *area of interest* of the user or to the context that the user give to the words.

The area of interest is computed using a breadth-first search, starting from all nodes associated to the keyword until an element of the *Profile* is found or until search has reached a *limit* depth[2] (which in practice is 5).

4.5 Step 4: Understanding

This step (understanding module in figure 2) shows to the user in a useful manner the set of shortest paths from the keyword to the profile classes. Since it is possible that the user wants to search out of the area of interest, the algorithm also shows the nodes directly connected to the keyword .

In case the resulting graph is too complex to be visualized, the graph is pruned using a transitivity rule. For example, if *a* is subclass of *b* and *b* is a subclass of *c*, the intermediate relation (*b*) is hidden and a direct arc between *a* and *c* is shown.

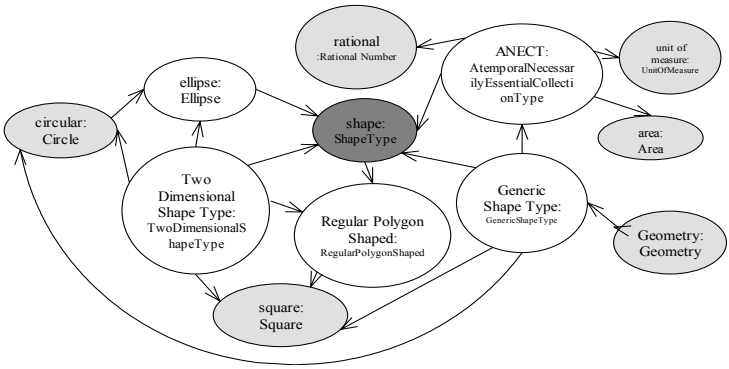

Fig. 7. Resulting Graph showed to the user. The dark gray node corresponds to the keyword and the light gray ones to the closest related elements in the profile.

[2] We have observed that at a greater distance, classes are generally in a context that has no relation with the search.

Now the system lets the user interact with the graph. The user can click over the nodes in the graph. Afterwards, the system shows all the web pages associated to the classes of the ontology clicked by the user. In our example, in case the user clicks the classes Shapetype and ellipse, the system will show links to the web pages that contain both concepts.

4.6 Step 5: Reflection

At this point (module reflection of figure 2), the system needs feedback from the user to verify that it has done the correct thing. If the user chooses one or more of the shown classes, it means that he agrees with what was displayed by the algorithm, that is to say, the concept he was looking for has been interpreted well by the system.

If the user chooses one of the nodes that do not belong to the paths between keyword and profile nodes, it means he does not agree with the result of the system. This can be because the concept was not interpreted well or the context found by the algorithm was not the correct one. At this point, an option is presented to the user in order to allow him to better customize his profile. The system offers him to add new concepts by changing the paragraph that defines his language.

5 Some Preliminary Tests

In order to test our system we compared it against the well-known search engine Google. The experiment consisted in searching for information about the Isle of Man (an country located in the Irish Sea).

We invoked Google with the keyword “Man”, and kept the first 44 results. From these, only 6 where related to the island (13.6% of efficacy).We processed these 44 pages with our system, generating a database that contained links to 742 different classes of the ontology. Afterwards, we invoked our system with the keyword “Man” (which in fact has three associated concepts in the ontology: “AdultMalePerson”, “ControllingSomething”, and ”Country”). Furthermore, we entered a profile with two classes: “Nation” and “GeopoliticalEntity”. In the resulting graph, when the user clicks over the node “man” he obtains a list of 18 links. Among them, 6 are related to the isle of Man (33% of efficacy).

In this simple test, we see that the efficacy of our prototype is better that of Google. Moreover, since our prototype “discovered” that what the user wanted was information of the isle of man, the information returned is fewer (only 18 from a total of 44 possible answers).

6 Conclusions and Future Work

The system presented is not a purely semantic web search engine, nor a syntactic one. The main advantage of our approach is that it can work with existent web technology, since it is not necessary to add any special context to HTML pages. Nevertheless, its main limitation is that it relies on the existence of a big ontology, containing, ideally,

all human commonsense knowledge. Although the existence of such an ontology could be regarded as an utopia, we think it is more promising that pure semantic web, since it is unlikely that common web authors will be able to (or want to) use advanced tools to semantically describe their pages.

One of the main contributions of this work corresponds to the use of a cognitive model of the way a human being retrieves information. This model has been enlightening to us, especially with regard to the design of the system.

Currently, we are doing extensive tests of efficacy and efficiency of the system. Furthermore, we plan to improve the efficiency of the observation step, replacing breath-first search by a fast algorithm that will use a database of pre-computed distances between ontology nodes.

References

1. Joaquin Delgado, Naohiro Ishii: Multi-Agent Learning in Recommender Systems for Information Filtering on the Internet. Int. J. Cooperative Inf. Syst. 10(1-2). (2001) 81-100
2. Gediminas Adomavicius, Alexander Tuzhilin: Using Data Mining Methods To Build Customer Profiles. Computer Innovative Technology For Computer Professionals, (2001) 74-82
3. William B. Frakes, Ricardo Baeza-Yates: Information Retrieval. Prentice Hall. (1992)
4. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern Information Retrieval, Addison Wesley. (1999)
5. Eduardo Mena, Vipul Kashyap, Amit P. Sheth, Arantza Illarramendi: OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. CoopIS 1996. (1996) 14-25
6. Perkins, D.N.: What Is Understanding? In M. S. Wiske (Ed.), Teaching For Understanding: Linking Research With Practice. San Francisco. (1998) 39-57
7. Van Heijst, G., Schereiber, A.T. Y Wielinga, B.J.: Using Explicit Ontologies In Kbs Development. International Journal of Human and Computer Studies (1996) 183-292
8. The DARPA Agent Markup Language Homepage, http://www.daml.org
9. Overview of OpenCyc, http://www.cyc.com/cyc/opencyc/overview
10. Appelt, D.:An Introduction to Information Extraction. Artificial Intelligence Communications, 12(3). (1999) 161–172
11. Atanas Kiryakov, Borislav Popov, Damyan Ognyanoff, Dimitar Manov, Angel Kirilov, Miroslav Goranov: Semantic Annotation, Indexing, and Retrieval. International Semantic Web Conference (ISWC). (2003) 484-499
12. Ontotext – a Sirma Lab for Knowledge and Language Engineering, http://www.ontotext.com/
13. Gardner, H.: Multiple Intelligences Approaches To Understanding. In The Disciplined Mind: What All Students Should Understand. Simon & Schuster, New York. (1999) 186-213
14. Gardner, H.: Perspectives Of Mind And Brain. In The Disciplined Mind: What All Students Should Understand. Simon & Schuster, New York. (1999) 60-85
15. Guilford, J.: The Nature Of Human Intelligence. Mcgraw-Hill, New York. (1967)
16. Perkins, D. N.: What Is Understanding? In M. S. Wiske (Ed.), Teaching For Understanding: Linking Research With Practice. San Francisco. (1998) (39-57)