A Semantic Search Engine for the International Relation Sector

L. Rodrigo[1], V.R. Benjamins[1], J. Contreras[1], D. Patón[1], D. Navarro[1], R. Salla[1], M. Blázquez[1], P. Tena[2], and I. Martos[2]

[1] Intelligent Software Components, S.A.
{lrodrigo, rbenjamins, jcontreras, dpaton, dnavarro, rsalla}@isoco.com
www.isoco.com
[2] Real Instituto Elcano
{pilar.tena, isabelmartos}@r-i-elcano.org
www.realinstitutoelcano.org

Abstract The Royal Institute Elcano[1] (Real Instituto Elcano) in Spain is a prestigious independent political institute whose mission is to comment on the political situation in the world focusing on its relation to Spain. As part of its dissemination strategy it operates a public website. In this paper we present and evaluate the application of a *semantic* search engine to improve access to the Institute's content: instead of retrieving documents based on user queries of keywords, the system accepts queries in natural language and returns answers rather than links to documents. Topics that will be discussed include ontology construction, automatic ontology population, semantic access through a natural language interface and a failure analysis.

1 Introduction

Worldwide there are several prestigious institutes that comment on the political situation in the world, such as the UK's *Royal Institute for International Affairs* (www.riia.org), or the Dutch *Institute for International Relations* (www.clingendael.nl). In Spain, the *Real Instituto Elcano* (Royal Institute Elcano, www.realinstitutoelcano.org) is fulfilling this role. The institute provides several types of written reports where they discuss the political situation in the world, with a focus on events relevant for Spain. The reports are organized in different categories, such as Economy, Defense, Society, Middle East, etc. In a special periodic report - the "Barometer of the Royal Institute Elcano" - the Institute comments on how the rest of the world views Spain in the political arena. Access to the content is provided by categorical navigation and a traditional full text search engine. While full text search engines are helpful instruments for information retrieval, in domains where relations are important, those techniques fall short. For instance, a keyword-based search engine will have a hard time to find the answer to a question such as: "Governments of which countries have a favorable attitude toward the US-led armed intervention in Iraq?" since the crux of answering this question resides in "understanding" the relation "has-favourable-attitude-toward".

[1] Juan Sebastián Elcano was a Spanish explorer, who commanded back home the first successful expedition to circumnavigate the globe in 1522.

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729 , pp. 1002 – 1015, 2005.

In this paper we present and evaluate a semantic search engine that accepts natural language questions to access content produced by the Institute.

In Section 2, we briefly describe the ontology of the International Relations domain. Section 3 details how we automatically populate the ontology with instances. Then, in Section 4, present the semantic search engine, and how we automatically establish relations between the Institute documents and the (instances of the) ontology. In Section 5, we provide a failure analysis of the system based on a test with unknown users. Finally, in Section 6 we provide conclusions.

2 An Ontology of International Affairs

When searching for a particular data, looking for a concrete answer to a precise question, a standard search engine that retrieves documents based on matching keywords falls short. First of all, it does not satisfy the primary need of the user, which is finding a well-defined data, and provides a collection of documents that the user must traverse, looking for the desired information. Besides, not all of the retrieved documents may contain the appropriate answer, and some of the documents that do contain it, may not be included in the collection. These drawbacks seem to suggest a change in the search paradigm, evolving from the extraction of whole documents, to the information contained in those documents. This approach, however, is not feasible in all conditions. It is not affordable to build such a search engine for general purpose, but only for limited, well-defined domains. This is the case of the semantic search engine developed for the Real Instituto Elcano, which focuses on the topics covered by the reports written by the institute analysts, this is, international politics.

In order to be able to analyse the documents, and reach the sufficient "understanding" of them to be able to answer the users questions, the system relies on a representation of the main concepts, their properties and the relations among them in the form of an ontology. This ontology provides the system with the necessary knowledge to understand the questions of the users, provide the answers, and associate it a set of documents that mention the concept of the answer. Based on the ontology, each document gets its relevant concepts annotated and linked to the representing concept or instance in the ontology, allowing a user to browse from a document to the information of a concept he is interested in, and backwards, from the ontology to any of the reports that mention that concept.

2.1 Ontology Design

An ontology is a shared and common understanding of some domain that can be communicated across people and computers [6, 7, 3, and 8]. Ontologies can therefore be shared and reused among different applications [5]. An ontology can be defined as a formal, explicit specification of a shared conceptualization [6, 3]. "Conceptualization" refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. "Explicit" means that the type of concepts used, and the constraints on their use are explicitly defined. "Formal" refers to the fact that the ontology should be machine-readable. "Shared" reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group. An ontology describes the subject matter using the notions of concepts, instances, relations, functions, and axioms. Concepts in the ontology are organized in taxonomies through which inheritance

mechanisms can be applied. It is our experience that especially the social part for building a commonly agreed ontology is not easy [2].

Based on interviews with experts of the Elcano Institute, we used the CIA world factbook (http://www.cia.gov/cia/publications/factbook/) as the basis for the design of the ontology of International Affairs. The CIA fact book is a large online repository with actual information on most countries of the world, along with relevant information in the fields of geography, politics, society, economics, etc.

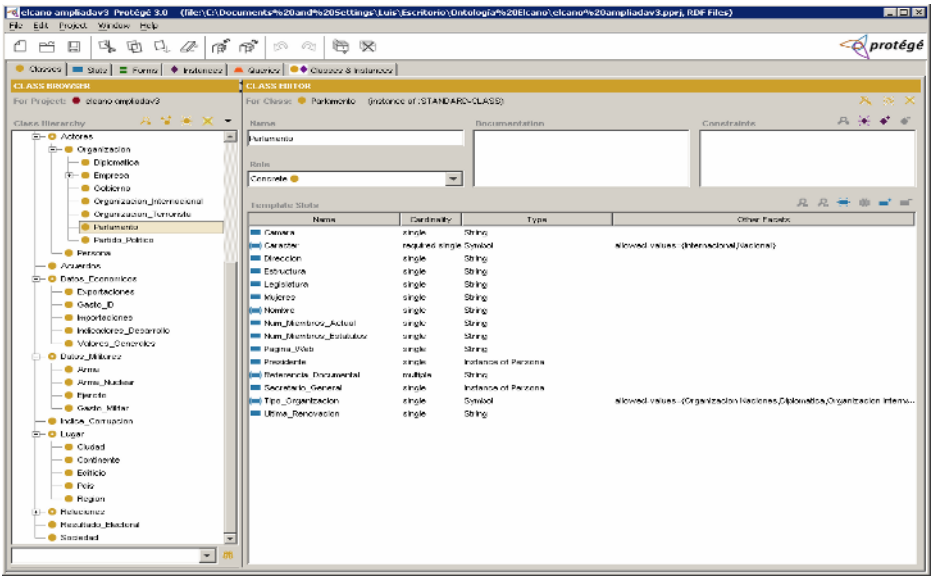

Fig. 1. Ontology for International Affairs

We have used the competency questions approach [10] to determine the scope and granularity of the domain ontology. The ontology consists of several top level classes, some of which are "*Place*" (representing geographical places such as countries, cities, buildings, etc.), "*Agent*" (extracted from WordNet [11], representing entities that can execute actions modifying the domain such as persons or organizations), "*Events*" (time expressions and events), and "*Relations*" (common class for any kind of relations between concepts).

Without instances information, the ontology contains about 85 concepts and 335 attributes (slots, properties). The ontology has been constructed using Protégé [9]. Fig. 1 shows a fragment of the ontology in Protégé.

3 Automatic Annotation

One of the challenges for the success of the Semantic Web is the availability of a critical mass of semantic content [17]. Semantic annotation tools play a crucial role at upgrading the actual web content into semantic content, that can be exploited by semantic applications. In this context we developed the Knowledge Parser ®, a system able to extract data from online sources populating specific domain ontologies, adding new or modifying existing knowledge

facts or instances. The Semantic Web community often calls this process as semantic annotation (or just annotation).

The Knowledge Parser ® offers a software platform that combines different technologies for information extraction, driven by extraction strategies that allow the optimal technology combination application to each source type based on the domain ontology definition.

Ontology population from unstructured sources can be considered as the problem of extracting information from the source, its assignation to the appropriate location in the ontology, and finally, its coherent insertion in the ontology. The first part deals with the information extraction and document interpretation issues. The second part deals with the information annotation, in the sense of adding semantics to the extracted information, according to domain information and pre-existing strategies. The last part is in charge of populating, i.e., inserting and consolidating the extracted knowledge into the domain ontology. The three phases can be seen in the architecture of the system, illustrated in Fig. 2.

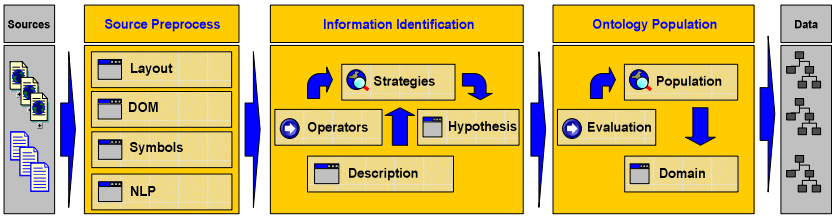

Fig. 2. Overview of the extraction and population process

3.1 Information Extraction

The KP system at present handles HTML pages, and there are plans to extend it to handle also PDF, RTF, and some other popular formats.

To be able to capture as much information as possible from the source document, KP analyzes it using four different processors, each one focusing on different aspects: the plain text processor, the layout processor, the HTML source processor an the natural language processor.

The plain text source interpretation supports the usage of regular expressions matching techniques. The usage of these kind of expressions constitutes an easy way of retrieving data in the case of stable, well known pages. If the page suffers frequent changes the regular expression becomes useless.

It is very common that even if documents of the same domain have very similar visual aspect they have a completely different internal code structure. Most of the online banks offer a position page where all the personal accounts and their balance are shown. These pages have very similar visual aspect, but their source code is completely different. The KP system includes layout interpretation of HTML sources, which allows to determine if certain pieces of information are visually located above or under, right or left, in column or in row, etc. of another piece of information.

In addition to HTML renderization of the source code in a visual model, the KP system needs to process the HTML elements in order to browse through the sources. The source description may include a statement that some information is a valid HTML link (e.g., a country name in a geopolitical portal), and when activated it drives to another document (a country description).

Finally, the fourth model tries to retrieve information from the texts present in the HTML pages. To do that, the user describes the pieces he is interested in in terms of linguistic properties and the relations among them (verbal or nominal phrases, coordinations, conjunctions, appositions, etc.)

3.2 Information Annotation

Once the document is parsed using different and complementary paradigms, there appears the challenge of assigning the extracted information piece to the correct place in the domain ontology. This task is called annotation, since it is equivalent to wrap up the information piece with the corresponding tag from the ontology schema.

The annotation of information is not direct in most of the cases. For instance, a numeric data extracted from the description of a country can be catalogued as the country population, the land area, or its number of unemployed people. It is necessary to have some extra information that allows reducing this ambiguity. This information, formulated in another model, enlarges the domain ontology with background knowledge, the same way the human use for its understanding. The extraction system needs to know, for example, that in online banking the account balance usually appears in the same visual row as the account number, or that the is usually followed by a currency symbol. This kind of information describing the pieces of information expected in the source and the relations among them is formalized in a, so called, *wrapping ontology*. This ontology supports the annotation process holding information describing the following elements: document types, information pieces and relations among the pieces (any kind of relation detectable by the text, layout, html or nlp models).

According to the domain ontology and the background information added, the system should construct possible assignments from the information extracted to the ontology schema. The result of this process is a set of hypotheses about data included in the source and their correspondence with the concepts, properties and relations in the domain ontology. During the construction process the system can evaluate how much the extracted information fits the information description.

The different ways in which hypothesis can be generated and evaluated are called strategies. Strategies are pluggable modules that according to the source description invoke operators. In the current version of the system there are two possible strategies available. For system usages where the response time is critical we use the greedy strategy. This strategy produces only one hypothesis per processed document using heuristics to solve possible ambiguities in data identification. On the other hand when quality of annotation is a priority and requirements on response time are less important we use a backtracking strategy. This strategy produces a whole set of hypothesis to be evaluated and populated into the domain ontology.

3.3 Ontology Population

The task of automatically filling a database or an ontological semantic model is non trivial, especially when the information comes from unstructured sources, where it may happen that the same information is repeated, spread over different places, or even inconsistent. For automatic ontology population, there is a need for a specialized module performing intelligent information integration tasks. In our architecture it is called IPO (Intelligent Population of Ontologies).

When the system has selected an information to be included in the ontology, there are different possible actions to take, which are: create a new instance with the information; insert found data into an existing instance; overwrite a value in an existing instance or, finally, relate two existing instances.

At this point, there is a key decision which affects the action to be taken: is the data found already present in the ontology, even under a different name? For that purposes, we have developed a library, SmartMatching, that decides whether two names refer to the same entity, and whether two instances in the ontology refer to the same concept in the real world. For example, it can decide that George W. Bush, Mr. Bush and Bush, G., all refer to the same person, and at the entity level, it can decide whether two entities holding economical data may be similar enough to suspect that they may belong to the same country (and therefore need to be unified into one single instance) or not.

The IPO module has to decide what hypothesis to insert into the domain ontology. Even if the input hypothesis is ordered from the most (the one that fulfils the source description with the highest degree) to the least probable, it does not take into account yet the consequences that introducing the data in the ontology may have. The best hypothesis may cause inconsistencies or may require many changes in the domain ontology. So there is a final decision step, in which one of the hypotheses generated in the previous step is populated in the ontology. For that reason the IPO makes simulations of the best ranked hypotheses and evaluates their suitability for final population according to their population cost. The population cost is calculated as the amount of changes needed in the domain ontology when filling new hypothesis. It means that hypothesis that contradicts and makes inconsistent the domain ontology has higher cost that the hypothesis that fits directly.

This way of disambiguation assumes that the information that is stored in the source intents to communicate something consistent and coherent with the information already stored in the domain ontology. On the base of the Shannon's information theory [Shannon] we understand that the online sources encode information looking to its easy understanding by the reader. The domain and the wrapping (description) ontology together try to reconstruct the possible mental model of the reader and guide the KP system in the task of its understanding. If the mental model is correct we assume that the cheapest interpretation (the one that require the smallest amount processing effort) is the correct one. We find here an application of the Occam's razor principle.

3.4 International Affairs Ontology Population

Using the Knowledge Parser system, we populated the ontology of international affairs, designed as described in Section 2.1. The domain experts selected four sources where they could find most of the information that they used on their daily basis. These four sources are:

- CIA World Factbook (http://www.cia.gov/cia/publications/factbook/).
- Nationmaster (http://www.nationmaster.com)
- Cidob (http://www.cidob.org/bios/castellano/indices/indices.htm).
- International Policy Institute for Counter-Terrorism (http://www.ict.org.il).

The set of sources is, of course, not exhaustive, but tries to follow the 80-20 rule, where a few sites cover most of the knowledge needed by the users of the system.

For each of the sites, a wrapping ontology was developed, describing the data contained in it, the way to detect it and the relations among them. The development of these kind of descriptive ontologies is at present done by experienced knowledge engineers considerably

fast, but it is in the plans for future advances to develop some kind of tools that will allow the domain experts to describe a new source and populate the ontology with its contents themselves. As a result of this process, we evolved from an empty ontology to an ontology with more than 60.000 facts.

4 The International Relations Portal

Modeling the domain in the form of an ontology is one of the most difficult and time consuming tasks in developing a semantic application, but an ontology itself is just a way of representing information and provides no added value for the user. What becomes really interesting for the user is the kind of applications (or features inside an application) that an ontology allows.

Following, we will present how we have exploited the semantic domain description, in the form of enhanced browsing of the already existing reports, and a semantic search engine integrated in the international relations portal, interconnected between them.

4.1 Establishing Links Between Ontology Instances and Elcano Documents

The portal holds two different representations for the same knowledge, the written reports from the institute analysts and the domain ontology, which are mutually independent. However, one representation can enrich the other, and vice versa. For example, an analyst looking for the GDP of a certain country may also be interested in reading some of the reports where this figure is mentioned, and, in the same way, someone who is reading an analysis about the situation in Latin America may want to find out the political parties present in the countries of the region.

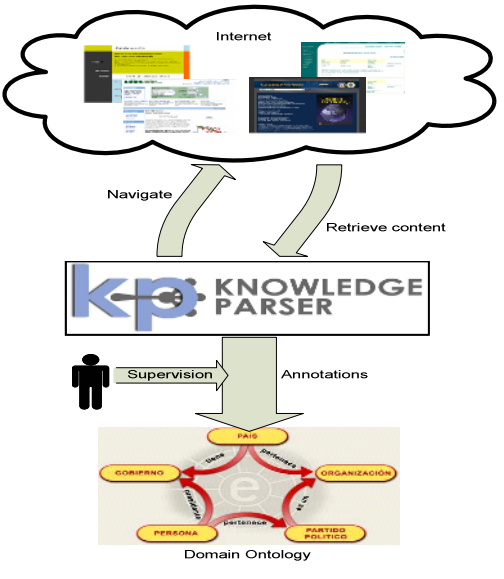

Fig. 3. Domain ontology population process

Trying to satisfy these interests, we inserted links between the instances in the ontology and the documents of the Institute. The links are established in both directions. Each concept in the ontology has links (references) to the documents where it is mentioned, and, viceversa, each document has links that connect every concept mentioned in the article with the corresponding concept in the ontology. This way, the user can make a question, for example, "¿Quién es el presidente de EEUU?" ("*Who is the USA president?*"), and gets the information of the instance in the ontology corresponding to George Bush. From this screen, he can follow the links to any of the instances appearing in the text, George Bush being one of them. This process can be seen in Figure 4, where the information about George Bush in the ontology contains a set of links, and the document seen can be reached following one of them.

To generate these links a batch process is launched, that generates at the same time both the links in the ontology and the links in the articles.

At present, the process of adding links is a batch process that opens a document, and looks for appearances of the name of any of the instances of the ontology in that text. For any matching, it adds a link in the text that takes to the instance in the ontology and link in the ontology with a pointer to the text. To evaluate the matching, not only the exact name of the instance is used, but also the possible synonyms, contained in an external thesaurus, which can be easily extended by any user, i.e., the domain expert.

Future plans include the automation of this task, so that any new document in the system (the institute produces new reports on a daily basis) is processed automatically by the link generator tool and the new links are transparently included in the system.

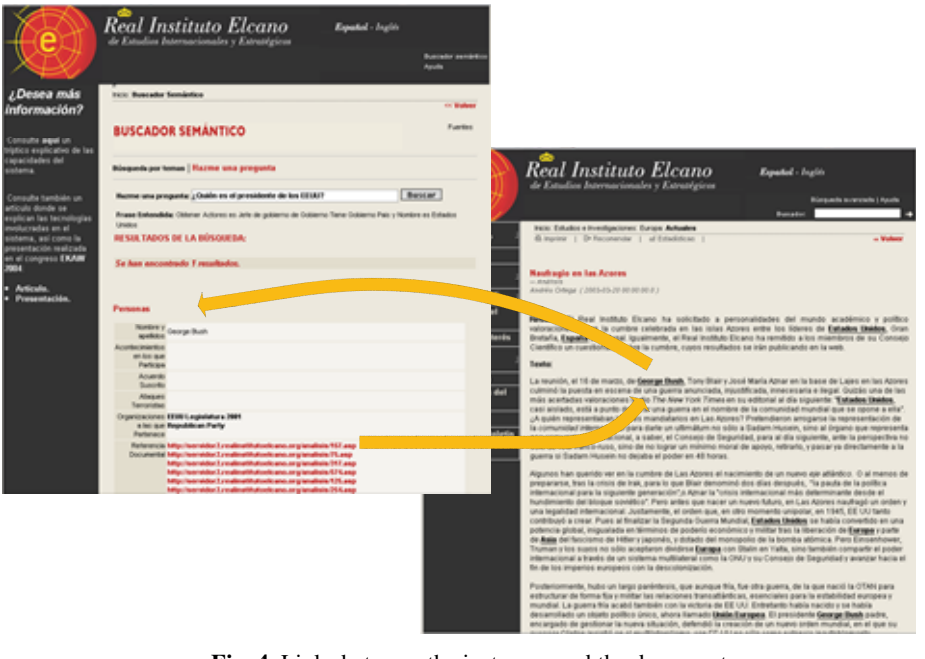

Fig. 4. Links between the instances and the documents

4.2 The Semantic Search Engine

With the objective of making available to the users the knowledge contained in the ontology in a comfortable, easy to use fashion, we also designed a semantic search engine. Using this engine, users can ask in natural language (Spanish, in this case) for a concrete data, and the system retrieves the data from the ontology and presents the results to the user.
The general steps that the system carries out with every query are the following:

· First of all, the user question is interpreted, extracting the relevant concepts from the sentence.
· Second, that set of concepts is used to build a query that is launched against the ontology.
· Finally, the results are presented to the user.

Each of these steps will further detailed in the following sections.

4.2.1 Sentence Analysis

When users go to the web searching for data, they expect a fast, almost immediate answer. If the process takes too long, the system just gives an impression of being unusable and the user will try with an alternative search engine. This is a serious drawback for these kind of systems that require heavy processing before being able to offer an answer to the user. Therefore, the process of obtaining an interpretation of the sentence of the user is optimized trying to provide an answer as fast as possible, sacrificing somehow the depth of the processing.

The module is organized as a cascade of modules, from the most simple to the most complex ones, and each time a module is executed, the system checks if it is necessary to go on with the following modules, or if an answer can be already delivered. This process can be seen in Fig. 5.

The input sentence is tokenized, obtaining a list of words, numbers, and punctuation signs. This list will be the input to the cascade of modules.

The first module detects and marks the words that do not contribute any relevant meaning to the sentence (known as *stopwords*), so that from the very first moment those words are ignored by the rest of the modules.

After every module execution, the system checks if the information collected is enough to provide an answer. To take this decision, the system checks if every token in the sentence is either marked as a stopword, as a punctuation sign or has any semantic information attached. If any of the tokens of the sentence is not annotated, the processing continues with the next module.

The second module, the first one that attaches semantic information, uses the ontology as a gazetteer. Basically, it goes through all the names in the ontology (names of classes, attributes, symbols and instances), taking also into account the synonyms files that were afore mentioned, and checks if any of them appear in the sentence. If so, it attaches to the word the information of the element of the ontology it represents. This information depends on which kind of element was recognized. If it was the name of a class, just that name is enough, while if the word matched the value of an attribute, the system attaches the name of the class, attribute and exact value in the ontology (the matching does not need to be exact, especially due to capitalization and grammatical accents).

The next module uses some shallow linguistic techniques to broaden the detection possibilities. The first thing that the system checks is if any of the words are operators that need

to be included in the query. At this moment, only negative (*no*) and comparative operators (*mayor que*, *menor que*, *igual*) are implemented, but future plans include temporal operators also. If this does not complete the sentence analysis, the system verifies if any of the tokens that could not be analyzed is a number, and if so, marks it as such. In the last step of this module, as the word in the sentence that could not be recognized does not match any of the terms contained in the ontology (it was checked in the previous step), the system looks for variations of the word. First of all, it tries with the lemmas of the words in the sentence. This is of special interest in a highly flexible language as Spanish when dealing with verbal forms. Finally, the last step is to check if any of the words that still have not been understood may have been misspelled. For this purpose, we have adapted a spelling corrector, *ispell* [20], adding to the corrector dictionary all the vocabulary contained in the ontology, so that it is optimized for the application domain. The corrections suggested by ispell are checked by the second module, to see if they appear in the ontology and, if so, the word is considered misspelled and the appropriate information is attached.

Finally, if there is still some token that has no information one last "desperate" process is launched that, using regular expressions, checks if the word is part of the name of any element of the ontology. This is quite helpful, as names in the ontology (particularly proper names) are written in their full form, while they are usually referred in a shorter way, for example, users will tipically write "Washington" while the name of the city in the ontology is "Washington D.C.".

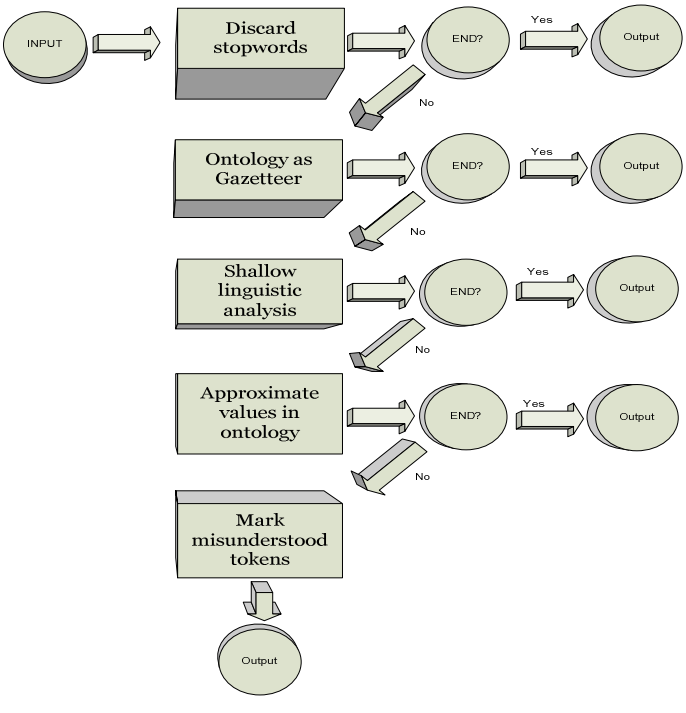

Fig. 5. Steps in the sentence analysis

If all the modules have been executed and any token remains unannotated, the token is marked as misunderstood, and the analysis finishes, returning the tokenized sentence with the corresponding information attached. Every token in the user question, therefore, will have some kind of information attached, which may range from a "Stopword" tag, a "Not understood" tag, or (hopefully) the semantic information that the system could build for it.

4.2.2 Quezry Construction

While human language is ambiguous most of the times, not mentioning facts that the speaker and the listener share, an ontology query language needs to explicitly mention all the facts that take part in the query. From the sentence analysis phase, the system gets a set of spots in the ontology, some classes, attributes and instances that the user has mentioned, but that is not enough to build a query. It is necessary to guess what kind of relations hold between those elements (that the speaker did not mention), so that the query can be well constructed. To achieve this, the system looks for the paths inside the ontology that connect all the elements mentioned in the sentence. Once a full path is found, a query can be constructed.

The process that is carried out at present to calculate the path does not consider possible permutations in the order of the tokens in the sentence, and calculates the minimum distance from one element to the next, not taking into account minimum global paths. This is not the optimal algorithm, but once again efficiency has been preferred to efficacy. Nevertheless, improvement plans include this module as one of the main options, as sometimes the algorithm is just too rigid, and even though the system may have been able to understand the user sentence, the query cannot be correctly built and the system fails to give a response.

Fig. 6. Results presentation

4.2.3 Results Presentation

Once the system has got one or more URIs that represent the result to the question, the information contained by these URIs is presented to the user as tables with the information contained in the ontology. An example of a visualization of results can be seen in Fig. 6.

5 Failure Analysis

The system has been tested during one month in the context of the W3C Spanish standards tour[2]. The scenario for this external evaluation was the following. Users were given a brief introduction to the system capabilities and functions, and then they could freely interact with it. One hundred utterances were collected from unknown, spontaneous users.

Summarizing the general processing of the system, depicted in Section 4, the system first tries to understand the natural language, translating it to an internal representation based on the ontology, and then tries to build a query that retrieves the instances from the ontology that satisfy the user question. These instances are finally presented to the user as an answer. If the system is not able to complete any of these two steps, it will not be able to offer a response. If this happens, the user will be given the option to redirect the search to an standard search engine, and, moreover, if the system detects that some words were not properly understood it also notifies the user about the problem and gives him the possibility to rephrase the sentence with new words.

The two possible sources for preventing the system from giving an answer have been explicitly identified in the results table, to be able to point out the one that is responsible for a greater number of errors.

Finally, answers classified as "Wrong result" denote an error in the system. It has been able to process the question and thinks that it is providing a sensible answer, but the answer does not correspond to the question. This kind of malfunctioning comes from design or implementation bugs which can be located at any level, in the ontology, in the sentence analysis, or in the query construction and should be studied individually to uncover the reasons.

Table 1. Evaluation figures

	Correct	**No result**		**Wrong result**	**TOTAL**
		NL error	***Query generation error***		
All Sentences	46	24	21	9	100
Domain sentences	46 (63.01%)	3 (4.11%)	17 (23.28%)	7 (9.59%)	73

From the figures in Table 1, we can conclude two main points. The search engine is clearly domain specific, and only if users understand the implications of this fact will they be able to use the system successfully. This is suggested by the dramatic decrease of errors (specially in the NL) when only domain specific sentences are

[2] http://www.w3c.es/gira/info/intro.html.en

considered. Additionally, this decrease suggest that the sources for data acquisition, that implicitly define the domain of the engine, should be chosen with great care and in agreement with the domain experts. The more sources that can be added, the more robust the system will behave.

We can also conclude that the second phase of the analysis, the query construction is at present the weakest link in the chain as it is not flexible enough to ensure that every well understood question will be correctly converted to the appropriate query. This point constitutes one of the future lines of improvements, and we will focus on it in the short term.

6 Related Work

Our Knowledge Parser is related to several other initiatives in the area of automatic annotation for the Semantic Web, including KIM [12], which is based on GATE [13], Annotea [14] of W3C., Amilcare [15] of Sheffield University, and AeroDAML [16]. For an overview of those approaches and others, see [4]. All approaches use NLP as an important factor to extract semantic information. Our approach is innovative in the sense that it combines four different techniques for Information Extraction in a generic, scalable and open architecture. The state of the art of most of these approaches is still not mature enough (few commercial deployments) to provide concrete comparison in terms of performance and memory requirements.

7 Conclusions

A semantic search engine for a closed domain was presented. The figures of the evaluation are promising, as more than 60% of the spontaneous questions are understood and correctly answered when these belong to the application domain. However, there are still some things to improve, such as the automatic link generation, a more flexible mechanism for building queries, an automated process to generate complete synonym files from linguistic resources, just to mention a few of them. It would also be of a high interest to completely decouple the search engine from the domain information, which are now lightly connected, in order to be able to apply the semantic search engine to a new domain just by replacing the domain ontology and the synonyms files.

The semantic search engine is, at the same time, a proof of the utility and applicability of the Knowledge Parser ® which will also be further developed in future projects.

Acknowledgements

Part of this work has been funded by the European Commission in the context of the project Esperonto Services IST-2001-34373, SWWS IST-2001-37134, SEKT IST-2003-506826 and by the Spanish government in the scope of the project: Buscador Semántico, Real Instituto Elcano (PROFIT 2003, TIC). The natural language software used in this application is licensed from Bitext (www.bitext.com). For ontology management we use JENA libraries from HP Labs (http://www.hpl.hp.com/semweb) and Sesame (http://www.openrdf.org/).

References

1. Gómez-Pérez A, et al (2003) Ontological Engineering. Springer-Verlag. London, UK.
2. V. R. Benjamins, et al. (KA)2: Building ontologies for the internet: a mid term report. International Journal of Human-Computer Studies, 51(3):687–712, 1999.
3. W. N. Borst. Construction of Engineering Ontologies. PhD thesis, University of Twente, 1997.
4. Contreras et al. D31: Annotation Tools and Services, Esperonto Project: www.esperonto.net
5. A. Farquhar, et al. The ontolingua server: a tool for collaborative ontology construction. International Journal of Human-Computer Studies, 46(6):707–728, June 1997.
6. T. R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199–220, 1993.
7. N. Guarino. Formal ontology, conceptual analysis and knowledge representation. International Journal of Human-Computer Studies, 43(5/6):625–640, 1995. Special issue on The Role of Formal Ontology in the Information Technology.
8. G. van Heijst, et al. Using explicit ontologies in KBS development. International Journal of Human-Computer Studies, 46(2/3):183–292, 1997.
9. Protege 2000 tool: http://protege.stanford.edu
10. M. Uschold and M. Gruninger. Ontologies: principles, methods, and applications. Knowledge Engineering Review, 11(2):93–155, 1996.
11. WordNet: http://www.cogsci.princeton.edu/~wn/
12. Atanas Kiryakov, et al. Semantic Annotation, Indexing, and Retrieval 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Florida, USA. LNAI Vol. 2870, pp. 484-499, Springer-Verlag Berlin Heidelberg 2003
13. H. Cunningham, et al. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002
14. José Kahan, et al, Annotea: An Open RDF Infrastructure for Shared Web Annotations, in Proc. of the WWW10 International Conference, Hong Kong, May 2001.
15. Fabio Ciravegna: "(LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts" in Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining, held in conjunction with the 17th International Conference on Artificial Intelligence (IJCAI-01), Seattle, August, 2001
16. P. Kogut and W. Holmes, "AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages", in Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001).
17. Benjamins, V., et al. Six Challenges for the Semantic Web. White Paper, April 2002.