JaBot: a multilingual Java-based intelligent agent for Web sites

Tim READ & Elena BARCENA
Departamento de Filologías Extranjeras y sus Lingüísticas, UNED
Senda del Rey s/n, Madrid 28040, Spain
timread@sr.uned.es, ebarcena@sr.uned.es

Abstract

This paper presents a novel type of intelligent agent with a multilingual natural language interface, which retrieves information from within a Web site. This agent, named JaBot after the fact that it is a bot which has been programmed in Java, has been designed and developed by the authors in an attempt to solve common Web site problems related to information retrieval. JaBot runs quickly and efficiently, and rather than running directly on the Web site pages, it is connected to a lexical semantic map. This map is based upon the contents of the Web site in question together with other associated linguistic knowledge.

Introduction

Java was launched by Sun Microsystems in the early '90s as a simple, robust, dynamic, multi-threaded, general-purpose, object-oriented, platform independent programming language! Its strengths can be split into four key issues, namely, portability, security, robustness and ease of usage, and distributed operation across the Web (Read et al., 1997). These benefits make Java an ideal programming language for constructing Web-based computational linguistic applications and agents (Ritchey, 1995; Sommers, 1997). Some applications of this type are beginning to appear on the Web, such as the English learning tools developed in Java by the authors as part of the UNED – Profesor Virtual (UPV) research project[1] (Read & Bárcena, in prep.).

[1] Although JaBot and the rest of the modules that make up the UPV are fully functional and have been operational for some time now locally on our departmental Web pages, they cannot be accessed yet on the Internet because our Web site is in the final stages of construction.

Access to the vast amounts of information contained on the Web still highlights some problems, such as that of cataloguing or indexing all that information. The sheer size of the Web and the ever changing nature of its contents means that the process of charting it is closer to mapping a large cavern with only the aid of a small torch than to the construction of a library catalogue.

Bot or agent technology is playing an increasingly important role in this mapping process, as will be seen next.

1 Bots and the Web

Bots are distinguished from other commonly used programs in that they act as if they have some degree of intelligence and independence (Thompson, 1998). Born in the '60s, nowadays bots should be viewed as part of the wider move towards distributed object-based systems (Weber, 1997). Instead of having massive programs, the tendency is to use networked computer systems made of a large number of co-operating task-specific components. Some of these components will act when told to; others, bots, will be more autonomous, making the on-line experience more pleasant and productive.

Internet search engines have a reputation of being unfriendly and unhelpful, despite the fact that some of them offer basic natural language interaction. The problem arises exactly at the point when the user connects to a specific Web site in search of some information that s/he believes to be contained there. If the site is large and there is no search engine, finding a particular item can be very difficult and time consuming, especially over a slow connection. Even if a search engine does exist, the current basis of search technology on the use of 'wild card'-based literal strings means that, unless the user knows a keyword which will be part of the entry s/he wants, the results of the search may well be zero links or a large list of

marginally related references in which the desired link is embedded.

In order to overcome these problems, the authors have designed and developed a bot which functions within a Web site.

2 JaBot - The design

In this section the design requirements and specification of the bot which has been devised and developed by the authors for searching within Web sites are presented. Its name is JaBot, which comes from 'Java-Based Bot': the word 'bot' in turn comes from 'robot', both of which are alternative words for 'intelligent agent'; and Java is the programming language in which the bot was written. There are four specific requirements which have driven the research and development process.

Firstly, a Web site assistant bot is required to facilitate the exploration of the contents of a site beyond the strict, limited manipulation of literal text chunks in blind searches. Given the lack of one-to-one correspondence between conceptual and linguistic units, the bot should be flexible in the sense that it should retrieve matches not just by using the input words literally, but rather by trying to "understand" the concept which concerns the user, so that the bot can search for the same information under different but semantically similar terms if necessary.

Secondly, reflecting this search flexibility, the interface to the bot should be in plain natural language, enabling questions to be presented in a natural way. Such an assistant bot would resemble the help system on Microsoft Office97 in the sense that questions here can be formulated in natural language and answered in terms of links within the Web site which relate to the subject of the question, i.e., its semantic content beyond the literal text it contains.

Thirdly, the interface should be multilingual so that users do not have to pose the query in the language of the Web page. Even though the users may not understand this language, their ability to formulate questions in their own language would enable them to, for example, access the details of a particular person (their telephone number or e-mail address) who may well speak their language.

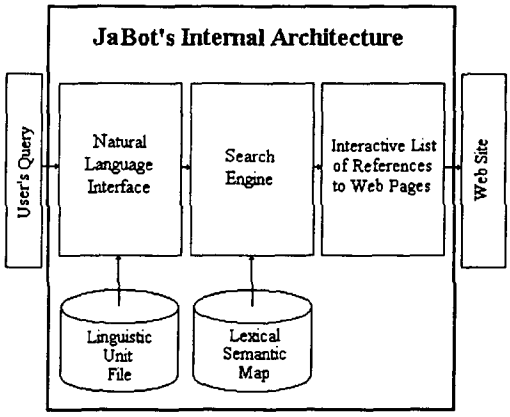
Fourthly and finally, the binary file which corresponds to the bot needs to be sufficiently small so that it can be transferred across the Internet at a reasonable speed. The tacit law of the Web is clear: if users have to wait too long for the bot to start working, it will not be used. This requirement has implications for the degree of sophistication of the linguistic processing and the types of data files associated with it.

Now that the requirements have been presented, the resulting design is described. JaBot is domain-specific in the sense that it can only operate on the Web site for which it was configured. This is useful from a practical functional perspective because it limits both the conceptual and linguistic diversity which needs to be processed (so far this approach has produced the best results in computational linguistic applications [Boitet, 1990]). In other words, users of JaBot will be formulating questions which attempt to locate information that is likely to be contained on the Web site, and not the full range of questions that they might like to ask a human expert on the subject. For example, if JaBot were placed on the Web site of a university department, users would be enquiring about subject contents, tutorial hours, exam dates, etc., and not attempting to ask which of subjects X and Y is easier or more relevant for their careers.

As can be seen in the diagram below, JaBot has three modules: a natural language interface, a search engine and an interactive list of references to the Web pages on the site at which it is operating. At start up time, two data files are loaded, namely, a file of linguistic units with little or no semantic relevance in the context of Web site information retrieval, and a lexical semantic map of the particular Web site. The linguistic unit file contains a list of the grammatical and lexical elements, marks, words and other literal strings which are not used when locating entries within the Web site. The lexical semantic map contains lexical elements (e.g., terms and compounds)

which correspond to the concepts extracted from the Web pages on the site, as well as other synonyms and quasi-synonyms which may be used to refer to them.

The construction of the linguistic unit file represents less of a problem than that of the lexical semantic map, since for a particular language the semantically empty elements will remain constant independently of the content of the Web site. Hence, once versions of this data file are constructed for the main languages used on the Web, they could be made publicly available for all sites. Both the linguistic unit file and the lexical semantic map have been formulated from an empirical study carried out by the authors on the way in which questions are typically asked about Web site contents.



The lexical semantic knowledge to be used by the search engine is extracted from the user's questions by a process of rudimentary parsing based upon the restrictions imposed by the linguistic unit file. In essence, the majority of the grammatical words and certain other literal chunks of language are removed leaving a string of key lexemes which belong to open linguistic categories. The parser does not take into account the punctuation of the query since it is assumed that the user has posed one single question, and not a series of questions or sentences with other communicative functions. This procedure is motivated by the fact that the grammaticality of such electronic input is often very low since it is closer to oral interactions than to carefully produced written texts (cf. the quality of e-mail).

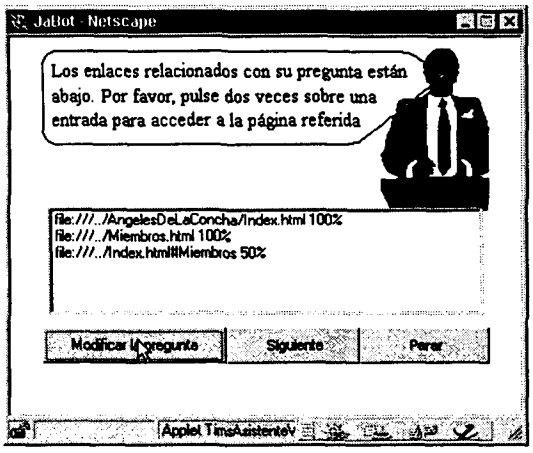
The remaining lexical elements are used by the search engine, not directly on the Web site, but against the nodes of the lexical semantic map. Each node in the map consists of a link to a Web page (or section) and a list of semantically similar words and expressions in the given domain. The links to the Web pages which correspond to the nodes of the map that have been activated in the search are presented to the user as a list, ordered by the number of elements found in each node. Double clicking on a link will retrieve the information by opening the corresponding page in the main browser window.

Finally, the multilinguality of JaBot depends on the way in which the lexical semantic map and the linguistic unit file are coded. If foreign language knowledge is included in both sources, then foreign language queries are possible. The content of the Web site (and therefore the responses to the user) would, however, not be multilingual unless the site had been constructed that way.

3 JaBot - A working example

The example presented here has been extracted from our Web site locally. JaBot contains a scrolling set of images which inform the user of its functionality, and also a text window into which the user can enter his/her questions, as shown in this diagram.

In this example someone wants to know who is the head of the department, and consequently enters the question: "Who is the head of department?". Such a question would produce the following output list of links:



Double clicking on the top entry will access the head of department's home page. When the way in which this question can be expressed in Spanish is considered, the advantage of JaBot over a simple literal string search engine (for example, the search tool which Microsoft FrontPage provides for Web sites) becomes evident. Typically the head of department would be referred to as: "el director / la directora", depending on the gender of the person.

Now, since the head of our department is a woman, a user accessing the site who does not know this would use the default gender in Spanish, which is masculine, and enter "el director". A literal string search would not be able to identify the relevant link. Furthermore, if the user does not speak Spanish very well and enters a synonym such as "jefe", "cabeza", "presidente", "el que manda", etc., s/he would not be able to locate the desired reference either. Since JaBot uses semantic associations, it will find the same references for sentences which include any of the above entries, as well as similar ones in English.

4 JaBot - The next version

Any future version of JaBot will need to improve its competence in two aspects: its linguistic sophistication and its knowledge location and retrieval capabilities. Firstly, the linguistic issues. JaBot contains relatively little linguistic sophistication. Input questions are semantically parsed in a way that enables JaBot to answer a large range of basic queries about a Web site with some degree of flexibility. However, the parser cannot distinguish between such requests as:

(a) "I want to know the phone numbers of the lecturers of Linguistics X and Y".

(b) "I want to know the phone numbers of all the lecturers of Linguistics except X and Y"

The parser's sensitivity to such grammatical words as "except" and "not" would expand the range of query sentences which JaBot could handle effectively. Also, the identification of conjunctions like "and" and punctuation signs like the full stop would allow multiple queries. Even sentential order could, in principle, be taken into account. However, there is a well known trade-off between theoretical linguistic sophistication and practical performance which is applicable here (Hutchins & Somers, 1992). While sentences (a) and (b) pose a linguistic problem for JaBot, they may not pose a practical one, since our study of the types of questions which users actually ask did not include a single example of this type.

In order to cope with complex, ambiguous and incomplete input, the next version of JaBot should be able to assess the quality of its own parsing and searching, so that it can request clarification from the user when necessary. On a practical note, a semi-automatic tool for preparing the lexical semantic map would be a great help for Web masters who are considering employing JaBot on their sites. Otherwise, the manual preparation of this file can be time consuming and, furthermore, it would be more laborious to keep the file up to date as the Web site changes.

Secondly, the knowledge location and retrieval issues. At the simplest level an agent is a piece of software whose primary task is to increase productivity through automation. Some agents, "intelligent agents", seem to have certain autonomy or do something which can be considered to be "smart" (such as determining the importance of a piece of e-mail by scanning it for words like "deadline" or "won the lottery"). JaBot's intelligence is limited. It can only answer questions about the content of the site. It cannot compare, deduce, guess, etc.

Furthermore, agents, whether intelligent or not, are either static or mobile. The former can only operate within the confines of a single machine or address space. The latter have been defined in formal terms as "objects that have behaviour, state, and location" (Sommers, 1997, p.3). They can move about the network, executing tasks at different places and interacting with other agents when necessary.

JaBot is currently a static agent in the sense that it can only access information on the Web site where it is located. However, research has been done by engineers at IBM on mobile Java agents, named aglets (IBM, www.trl.ibm.co.jp/aglets/), which are able to move between Web sites running the aglet server. This mobility enables interaction between the aglets, which can be used to facilitate many different forms of behaviour, such as the sharing of expertise and information. Hence, a future version of JaBot could be designed as an aglet, which would enable it to continue functioning as it does at the moment on the local Web site, but with the additional capability to leave the site and interact with other JaBot aglets on servers where other related information is located.

A JaBot aglet may, for example, exist on the Web pages of the different departments of a university (located on physically different machines). Where user questions go beyond the information which is held on a particular departmental server, the JaBot aglet could leave its own server and go and interact with another one located elsewhere. Such mobility and the functionality which it entails may be very useful, for example, in the case of a modular degree where a student has to study courses in different departments and therefore wants to ask questions which relate to more than one area of knowledge.

Conclusion

In this article the problems which exist in the retrieval of information from a Web site have been considered together with the way in which a bot could be used to improve the situation. JaBot, a Java-based bot, has been designed and developed by the authors to overcome such problems. A requirements analysis has been undertaken, followed by the resulting specification of its architecture and associated data sources. Subsequently, an illustrative example of its functionality has been presented, which demonstrated that JaBot is more flexible than a traditional literal string-based search tool (where one exists). Other benefits of JaBot have also been identified, such as the way in which desired information can be accessed on the site without the need to know exact key words which exist in the entry. Furthermore, its ability to process questions in languages other than that in which the Web site was written. Finally, some limitations in the current design of JaBot have been outlined together with an indication of the form that the next version of this bot will take.

References

Boitet C. (1990) Towards Personal MT: general design, dialogue structure, potential role of speech. In H. Karlgren (ed.) *COLING-90: Papers presented to the 13th International Conference on Computational Linguistics* (**3**), pp. 30-35.

Hutchins W.J. and Somers H.L. (1992) *An Introduction to Machine Translation.* Cambridge University Press.

Read T. and Bárcena E. (in prep.) Cómo se prepara el Departamento de Filologías Extranjeras y sus Lingüísticas para el siglo XXI. *Revista de la UNED.*

Read T., Bárcena E. and Faber P. (1997) Java and its role in Natural Language Processing and Machine Translation. In *Proceedings of the Machine Translation Summit VI.* pp.224-231.

Ritchey T. (1995) *Programming with Java!* New Riders.

Sommers B. (1997) Agents: Not just for Bond anymore. *JavaWorld* (Electronic magazine at www.javaworld.com/jw-04-1997/jw-04-agents.html).

Thompson B. (1998) It's a tough job but somebot's got to do it. *Internet Magazine.* pp.44-48.

Weber J. (1997) *Using Java 1.1.* Que.