Information Exploration Using Mobile Agents

JAWAD BERRI and MOHAMMED AL-KHAMIS
Department of Computer Engineering
Etisalat College of Engineering
P.O. Box 980 Sharjah
UNITED ARAB EMIRATES

Abstract: - This paper presents a distributed knowledge-based system that explores sentences containing specific term related types of information from a network. The system uses a linguistic knowledge base including linguistic patterns and word lists representing expressions of the language for a specific type of information. The system is implemented with a mobile agent software environment in order to reduce the network traffic by moving the computations to the server side. It includes a set of processing tools which allow a specialized mobile agent to migrate in the network, process texts on the hosts, extracts relevant sentences that are matched to linguistic patterns and then returns the results to the client. The system has been used to search for information in a local area network.

The natural language engineering aspects of the approach, the system architecture and the implementation are discussed in this paper.

Key-Words: - Natural language engineering, contextual exploration, knowledge-based system, mobile agents, linguistic pattern, matching algorithm.

1 Introduction
The proliferation of textual information sources and their availability with the World Wide Web has opened the way to new research fields. The challenge is actually to obtain maximum benefit from the masses of information in order to provide services to different kinds of users. Searching information about terms is an application that can take advantage of the existing information sources. Terms are language words adapted from normal use or made up for a specialized usage within a given field of discourse. They are generally defined in the specialized field references but rarely cited in language dictionaries and lexicons. Moreover, for fields in continuous mutation such as Information Technology, finding specific information about newly produced terms is actually a tricky exercise. The difficulty is not the lack of information, especially with the Internet and the millions of WebPages retrieved, but it is actually to retrieve the piece of information that responds exactly to the user needs.

This paper addresses the abovementioned search issue and presents a system that retrieves specific types of information related to a term. A variety of information types can be explored with our system: definition, entailment, similarity, is part of, hyponymy, etc. The system is implemented with mobile agents. The user queries the system by typing the desired term and then specifies which type of information he wish to obtain. The system then creates a specialized mobile agent which is assigned a particular task. The mobile agent takes the necessary linguistic data and tools to achieve the requested task before it is dispatched into the network. It processes texts and matches the sentences with linguistic patterns. Upon completion, the mobile agent sends back the sentences containing the requested information.

In the next section we present the background of this work and the method used to extract sentences. Section three presents the knowledge acquisition process that aims at acquiring the necessary linguistic expertise. Section four presents the implementation of the system and the last section concludes this paper with a summary and suggestions for future work.

2 Background
Dealing with free text in heterogeneous information sources requires from a system to comply with three constraints: robustness, flexibility and adaptability. Robustness allows systems to solve problems even with incomplete linguistic data, which is the case for most problems related to language. Flexibility is concerned with the optimization of the efforts required to modify a system. This can be achieved by using software engineering paradigms that allow effortless updating of a system and potential reuse of its components in other applications. Adaptability is the ability for a system to be transportable to other
domains and to be scalable in order to handle large data. The latter constraint became a necessity to avoid the implementation of laboratory-restricted systems and to be able to process the heterogeneous amounts of free text available nowadays. Natural Language Engineering (NLE), a field at the cross roads of Natural Language Processing and Software Engineering, has emerged during the last decade in order to respond to the increased need of systems complying with the three above-mentioned constraints. NLE aims at providing solutions to real-life problems related to natural language with less cost and within reasonable time. NLE approaches usually make use of heuristics instead of large and complex combinatorial algorithms and rely on some language regularities that are observed in a corpus. NLE takes advantage of approaches and tools developed in software engineering and avoids a complete description of the problem [1].

To fulfill the increasing demand of NLE systems capable of providing rapid solutions to real-life problems, researchers have turned from the theoretical view of the problem towards more practical approaches. Contextual Exploration (CE) method developed by LALICC 1 research group is a promising approach since the analysis of texts focuses primarily on the linguistic knowledge and avoids a complete representation of texts [2]. The Message Understanding Conferences (MUC) claimed that the development of systems with a full syntactic analysis and a comprehensive semantic interpretation are not necessary for information extraction in a limited domain [3]. Answer Extraction from technical manuals can also be achieved without requiring a full semantic representation of technical texts. The system ExtAns uses a variable matching strategy to extract answers to user queries [4]. Knowledge acquisition from texts can also be done without a deep text analysis system. In [5], the authors present a system that combines linguistic and statistical approaches to acquire knowledge from texts. In the next section we present in detail the CE method that has been used in this work.

2.1 Contextual exploration

Contextual Exploration [6], [7], [2], is a decision-based method that is able to solve some classical computational linguistic problems with less processing cost. It has been implemented successfully to resolve classical linguistic problems such as tense and aspect [7], automatic summarization [8] and knowledge acquisition and modeling [9]. The method scans a context looking first for linguistic markers that can trigger decisions. Then the context is further analyzed to find linguistic contextual clues surrounding the marker to reinforce taking an unambiguous decision. CE method simulates the behavior of a reader who aims at taking a decision regarding a given linguistic problem. In the case of extracting definitions from texts, CE method can be applied by looking for a marker that is usually the verb. Then, the context (that is the sentence) is analyzed so that to find any clues (prepositions, determinants, adverbs, etc.) belonging to a specific word list. Finally the corresponding decision is taken and the sentence is tagged accordingly.

Contextual exploration method aims to build a knowledge-based system. Figure 1 shows the three analysis stages of CE. The first stage consists of the necessary linguistic data grouped into word lists to feed the knowledge base. Linguistic data is collected throughout an off-line process carried out in collaboration with the linguists. The objective is to identify and classify the word lists that are involved in taking the decisions. This stage is described in detail in Section 3. The second stage includes the linguistic patterns that are used to find specific information in the texts. Linguistic patterns are general schemas for which instances occur in the discourse in the form of expressions incorporating words from the word lists. Text word occurrences are matched with the word lists and tagged accordingly. The third stage includes a set of system’s decisions. In information exploration, decisions correspond to a semantic interpretation of

---

1 LALICC (LAngages Logiques Informatique, Cognition et Communication) is a research group of Paris-Sorbonne University in France.
the sentence content. A decision is taken whenever a set of word occurrences is matched to a linguistic pattern.

2.2 Linguistic patterns
For information exploration, linguistic patterns seem to be a reliable way to attain directly the meaning of sentences without the need of a whole description of the text sentences. Linguistic patterns are domain independent, they are used in languages to describe particular information. Their number is relatively limited hence efforts have to be deployed in order to gather these linguistic patterns and then employ them to extract the desired information. In the Section 3 a detailed acquisition of linguistic patterns is given. Similar work is done in knowledge-based information extraction to provide efficient solutions for text processing. The yearly Message Understanding Conferences involved several information extraction systems to extract specific information from texts. Most of these systems could interpret texts without requiring full syntactic and semantic analysis. Instead, they used various forms of linguistic patterns to interpret texts. For instance a pattern like [(perpetrator) attack (target) with (instrument)] can be matched to the sentence “Urban guerillas attacked the administration office with explosives” to extract information related to a terrorist event.

3 Linguistic knowledge acquisition
A major obstacle facing the development of robust systems that allows a user to query a collection of documents, process them, and extract relevant information is the need for large amounts of linguistic knowledge to handle the complexity of language analysis. Knowledge acquisition is the bottleneck for knowledge-based systems which analyze the language content.

Knowledge acquisition aims at transferring the problem solving expertise from an expert or some knowledge source to a program. Generally the transfer is accomplished by a series of interviews between a domain expert: the linguist, and a knowledge engineer who then writes a computer program representing the knowledge. Knowledge acquisition is done through a four steps cycle Elicitation, Representation, Implementation and Validation. Figure 2 depicts the whole process. Elicitation, consists of identifying and classifying the linguistic data and the defining models that can map it. This step has been carried out mainly by using the British National Corpus (BNC) and a set of texts selected randomly from the Internet. This process results in the identification of a set of linguistic patterns and the corresponding word lists. Representation consists of representing the knowledge in a formal language so that to be closer to the implementation. Then, the representation of the expertise must be turned into a runnable program. This is done in the implementation step where the expertise has been expressed into the JAVA language and implemented using Aglets. Testing is the last step where the expert has to test and verify the missing, incomplete or incorrect system data and rules.

3.1 Linguistic pattern acquisition
The expression of the definition in English can extremely vary in terms of the grammatical structure and the vocabulary used. Fortunately, the number of the expressions frequently used to define words is relatively limited. The first step towards defining linguistic patterns is to collect these expressions from corpus. This process has been done manually using the two main resources: the BNC and the Internet. Linguistic expressions are then grouped under common linguistic patterns. Table 1 shows some of the expressions collected and their corresponding patterns for the definition.

Table 1

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urban guerillas attacked the administration office with explosives</td>
<td>[(perpetrator) attack (target) with (instrument)]</td>
</tr>
</tbody>
</table>

2 The British National Corpus is a very large (over 100 million words) corpus of modern English. It is available at http://sara.natcorp.ox.ac.uk/
Table 1 Linguistic expressions for the definition and their corresponding patterns.

<table>
<thead>
<tr>
<th>Linguistic expressions</th>
<th>Linguistic Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>X is defined as Y</td>
<td>1) X + be + LVdefine + Las + Y</td>
</tr>
<tr>
<td>X is defined by Y</td>
<td>2) Y + LVpermit + to + LVdefine + X</td>
</tr>
<tr>
<td>X is defined in terms of Y</td>
<td>3) Y + LVuse + to + LVdefine + X</td>
</tr>
<tr>
<td>X can be described as Y</td>
<td>4) Y + LVcan + LVdefine + X + Las</td>
</tr>
<tr>
<td>Y allows to define X</td>
<td>5) Y + LVdefine + X + Las</td>
</tr>
<tr>
<td>Y permits to identify X</td>
<td>6) X + be + Lthe&amp;a + Y</td>
</tr>
</tbody>
</table>

The left column shows the expressions of the definition as found in corpus. The right column includes the linguistic patterns that result from the aggregation of the expressions.

### 3.2 Word list extension

Once the linguistic patterns and the word lists are defined, an important step is to extend the word lists so that to extend the coverage of the linguistic patterns. This step has been done using two main resources: WordNet and the Virtual Thesaurus.

Adding a word into a list must be carefully done. In general, synonyms are the first candidates to check. However, the set of synonyms as provided in dictionaries is too broad and encompasses the whole language. A fully automatic acquisition of synonyms from online dictionaries can give very odd results. This is why the extension operation is controlled by two constraints. The first is to consider the synset (synonym set); a set of words that are interchangeable in some contexts. Fortunately, the two thesauruses we are using provide this useful characteristic. The second constraint is to substitute in the original context any new word and then present it to the linguist who needs to validate any addition. Hence, word list extension is done mainly by contextual synonymy leaving the final decision to the expert.

Table 2 Word list extension.

<table>
<thead>
<tr>
<th>List name</th>
<th>Common Meaning</th>
<th>Word List</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVdefine</td>
<td>definition for the meaning of a word</td>
<td>define, specify, describe, determine, redefine, identify</td>
</tr>
<tr>
<td>Las</td>
<td>in terms of</td>
<td>as, by, in terms of</td>
</tr>
<tr>
<td>LVpermit</td>
<td>Make it possible to do something</td>
<td>allow, permit</td>
</tr>
<tr>
<td>LVuse</td>
<td>employ something for a particular purpose</td>
<td>use, utilize, employ</td>
</tr>
<tr>
<td>LVcan</td>
<td>Expresses the permission</td>
<td>can, may, might</td>
</tr>
<tr>
<td>Lthe&amp;a</td>
<td>the, a</td>
<td></td>
</tr>
</tbody>
</table>

The table shows the extended word lists which correspond to the linguistic patterns of table1. The first column is the list name as used in the system. The second column is the common meaning of the words included in the list. The third column presents the list words.

### 4 System implementation

#### 4.1 Mobile agents based implementation

Exploring information on a network is naturally a distributed application that relies on communication involving multiple interactions between the client and the server. As a result, lot of network traffic is generated especially on the client side. For this reason, we adopted a mobile agent environment to implement our system.

An implementation with mobile agents is an elegant solution mainly for two reasons: first, mobile agents reduce the need for the bandwidth. They allow all the client-server interactions to be packed together and dispatched to a destination host, where the interaction can take place locally. The motto is

---

3 WordNet is an online lexical reference system. It is available at [http://www.cogsci.princeton.edu/~wn/](http://www.cogsci.princeton.edu/~wn/)

4 Virtual thesaurus is a visual representation of the English language. It is available at [http://www.visualthesaurus.com/](http://www.visualthesaurus.com/)
simple: move the computations to the data rather than the data to the computations. Second, mobile agents take advantage of all the object-oriented paradigm features. They encapsulate all the required data within themselves. They may easily be dispatched and cloned in different directions. This allows them to function in parallel and cooperate to achieve a given task.

In addition, there are many other benefits for using mobile agent technology such as being able to adapt dynamically to a given situation, being naturally heterogeneous and able to react dynamically to unfavorable situations and events.

4.2 The Aglets workbench
IBM’s Aglets workbench\(^5\)\(^1\), \(^2\), is the software environment used to implement our system. The main reason of using such a system is to take advantage of the built-in mobility feature when dispatching agents in a network. Other reasons for this choice are: i) Aglets is purely based on Java programming language, it is therefore platform independent; ii) the package is free and easy to install; iii) the agent application contains a graphical user interface. Aglets workbench is one of the first complete Internet agent systems to be developed on the Java class library. Aglets (a pun on “agent” plus “applet”) are Java objects that can move from one host on the Internet to another. That is, an aglet that executes on one host can suddenly halt execution, dispatch itself to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its data. Aglets Workbench is a visual environment for building network-based applications that use mobile agents to search for, access, and manage corporate data and other information. Aglets workbench allows users to create mobile platform independent agents based on the Java programming language.

4.3 System architecture
Figure 3 illustrates the architecture of the system. When the user queries the system, a specialized mobile agent is automatically created and filled with the necessary data and processing tools to achieve its task. First, the mobile agent fetches from the linguistic database the necessary patterns and word lists which will be used to retrieve the requested type of information. Actually the system considers only definitions. However, current research work is undertaken to consider extending the linguistic database to other information types representing semantic relationships such as entailment, similarity, is part of and hyponymy. Depending on the nature of the task, the agent carries the necessary tools to process texts: the tokenizer (which segments the text into words and sentences), the stemmer (we are using the Java Porter stemmer \[^5\] ), the matching algorithm (see 4.3 for more detail), the scoring/ranking functions (that scores sentences based on how the patterns are matched), and the dispatching strategy defining how the agent will move on the network. The agent travels throughout the network and visits the hosts. It processes the texts and retrieves the sentences which are labeled by the linguistic patterns. Then, the agent returns to its sender and reports the sentences it retrieved with the corresponding file location and host address.

4.4 Matching algorithm
The matching algorithm matches linguistic patterns to sentences. It is completely decoupled from the data so that to allow the system to be updated easily. Hence, the linguistic patterns and the word lists can be updated without affecting the algorithm and vice-versa. The matching algorithm is implemented with three pattern matching options including the Pure Sequential, Sequential and Random search modes. These options were implemented in order to cope with the diversity of words to search for and to provide different search constraints which allow to loose or tighten the search depending on the search results. The Pure Sequential search mode represents the highest constraint since it forces the system to

\(^{5}\) Aglets Workbench is developed by the IBM Tokyo Research Laboratory.
match the words of the pattern with expressions in the text which appear in a strict consecutive order. Using this search mode no intermediate text tokens are allowed between the words of the patterns. For example, if we search for “mobile agent” definitions, the sentence “Mobile Agents are defined as software or hardware entities ...” is labeled as a definition whereas the sentence “Mobile Agents are defined in this paper as software or hardware entities ...” is not, since the words matched (underlined words in the sentence) are not consecutive to each other. The latter sentence is labeled with the Sequential search mode. Actually, this search mode requires that the matched words are in sequence but accepts the presence of other words inside the matched expression. The Random search mode necessitates the presence of all the words of the pattern in the sentence with no specific order. Furthermore, the algorithm considers the length of the pattern (number of words in the pattern) as another constraint. Actually, the patterns are ordered according to their lengths and the algorithm starts with the most restrictive patterns that are patterns with the highest length. If no result is found, the algorithm follows the Sequential search mode. Actually, this search mode requires that the matched words are in sequence but accepts the presence of other words inside the matched expression. The Random search mode necessitates the presence of all the words of the pattern in the sentence with no specific order.

4.5 Linguistic patterns and word list implementation

The implementation of linguistic data involves mainly two main JAVA classes: WordList and WordPattern. The former is mainly used as a container for a group of words belonging to the same word list. It is an object including the words of the list and a flag indicating whether or not to use stemming or not. The implementation of the WordList in the following program code:

```java
private static WordList LVdefine = new WordList("define", "specify", "describe", "determine", "redefine", "identify", true);
pdamite static WordList Las = new WordList("as", "by", "in terms of", false);
```

The WordPattern class is used to represent linguistic patterns which include one or more word lists. The `addlist()` method allows the creation of a pattern by integrating already existing word lists. Then, using the `addPattern()` method, the newly created pattern is added to a data structure that references all the system patterns. The Implementation is shown below:

```java
// let's build our PatternMatcher
pm = new PatternMatcher();
// create the WordPattern [X + be + LVdefine + Las + Y]
WordPattern pattern1 = new WordPattern();
// add the appropriate WordLists
pattern1.addList(verbbe);
pattern1.addList(LVdefine);
pattern1.addList(Las);
// the created WordPattern is added to a vector
pm.addPattern(pattern1);
```

Notice that the sequence of statements adding WordLists to the WordPattern must be in the order as defined by the linguistic expertise (see Table 1). This is mainly because the algorithm sticks to the prescribed order when matching sentences in both the pure sequential and the sequential modes.

Fig. 4 lists the definition sentences obtained for the term “mobile agents” with the sequential (normal) matching mode. Retrieved sentences are ranked according to the score they achieved. They are displayed along with their location (file and host). Note that the system highlights the sentence words that have been used in the matching.

![Fig. 4. Results for the query “mobile agents”](image)

The system is installed in a client machine and the server (Tahiti server) is installed in the host
machines. All the LAN computers contain a set of selected texts on various topics. On the client side, the user enters the term and selects the definition mobile agent to activate. The user selects also the search mode (pure sequential, sequential or random). The agent is then dispatched on the LAN from host to host, it searches for term definitions, extracts the relevant sentences and finally returns back with a list of ranked and scored results with details about the file and the host from where each sentence has been retrieved.

5 Conclusion
Exploring specific types of information from available text resources is an application that can considerably simplify users’ time and satisfy their particular information needs. In this paper, a system that extracts term definitions from texts was presented. The system uses the contextual exploration method to acquire the linguistic expertise including linguistic patterns and word lists. The system is implemented with mobile agents to reduce the network traffic by moving the computations to the server side. Therefore, when a specialized mobile agent is assigned a task, it takes from the system the necessary data and processing tools that allow it to extract the required information. The system has been used in a local area network to search for specific types of information from texts. The results obtained are encouraging.

Involving several specialized mobile agents looking for different types of information at the same time needs to be managed carefully. Actually, strategies to disseminate mobile agents in the network must be established in order to avoid conflicts and make the agents efficient in achieving their tasks.

References: