ABSTRACT
This paper describes an on-going Arabic Text Mining system. It presents the architecture of the system and provides an overview of some issues involved in Arabic Text Mining. This system uses techniques from the research areas of Data Mining and Natural Language Processing. The text analysis starts by a preprocessing stage to convert HTML Arabic documents to XML documents. This stage significantly simplifies the analysis process. The processed text is then linguistically analyzed from the word level to the text level. The result of this analysis is a semantic network of the entities mentioned in the text and the relationships among them. This semantic network is then used for some specific mining tasks.

KEYWORDS
Arabic Text Mining, Arabic Analysis, Natural Language Processing

1. INTRODUCTION
Database systems store information in the form of structured data and provide methods for querying them to obtain information that satisfies the user's query. Researchers in Knowledge Discovery Databases (KDD), also known as data mining (Witten 1999, Feldman 1998) have provided new techniques to extract implicit, previously unknown, and potentially useful information for particular purposes from structured databases (Ichimura 2001, Frawley 1991). Work in this area includes applying machine-learning and statistical-analysis techniques towards the automatic discovery of patterns in structured databases. However, in the past decade most efforts in KDD have focused on data mining from structured databases, despite the continuous growth of the volume of electronic unstructured data currently available on the Internet, Intranets, e-mail and GroupWare systems (Semio Corporation 1999). Most recently, text mining has emerged as new technology that aims at finding pattern in text documents. Compared to data mining which aims at finding patterns in structured data, text mining is more challenging as the analyzed data is unstructured. Text databases are rapidly increasing and becoming more available due to the increasing growth of information available in electronic format.

There has been little work on mining unstructured text (Feldman 1998). Rajman and Besancon (Rajman 2003) presented two examples of English text mining tasks: probabilistic associations of key-words and prototyping document instances. Ichimura et. al. (Ichimura 2001) proposed a text mining methods for analysis of a salesperson's daily reports.

While most of the studies on data and text mining have been conducted in English and other European languages, this paper focuses on the problem of Arabic text mining. Due to the huge amount of Arabic online information that appears in collections of unstructured text, Arabic text mining becomes necessary to make sense of all this information and to perform knowledge discovery from collections of unstructured Arabic text.

We argue here that Arabic is a very challenging language at all levels of analysis. Morphologically, Arabic words are very rich and carry a lot of lexical and grammatical information. Syntactically, Arabic sentence structure is different from that of English and other language. Semantically, Arabic words carry a lot of information that cannot be found in words of other languages.
In the next section we describe the architecture of our Arabic text mining system. Section 3 discusses the text pre-processing stage. Section 4 discusses the NLP module. Some concluding remarks and future work is given in Section 5.

2. SYSTEM ARCHITECTURE

Figure 1 shows the architecture of our Arabic text mining prototype system. This architecture is based on n-tier model. The client tier in this model is a Web browser which sends the user’s Arabic queries to the web server and displays the returned results back to the end user.

The middle tier, where most of the processing takes place, consists of a Web server, a scripting engine, a Natural Language Processing (NLP) module, an SQL query generation module, and a text mining module. The web server communicates with the client through the standard HTTP protocols, and communicates with the database server and the NLP module through the scripting language. The scripting engine is needed to handle server side processes. The NLP module is used to analyze the Arabic documents and queries.

The database tier consists of an SQL Server and the databases we need to query. This SQL server implements the client/server architecture which is important for web enabled applications.

In this paper, we will limit the discussion to the text preprocessing and the NLP stages. This is described in the next sections.

3. TEXT PREPROCESSING (HTML TO XML)

We have a collection of 42000 Arabic documents (316 MB), mostly from Al-Hayat Arabic newspaper. These documents are represented in HTML. Documents use HTML table to represent the news articles important fields like Author, Date, and Title. To provide better representation structures to these Arabic documents we have transformed them to XML documents. Figure 2 below shows the transformation process.
One of the problems posed by HTML documents is the inclusion of non-closed tags such as `<meta ...>`, `<br>`, `<p>` etc. Meta tag and some other tags are valid open tags in HTML and can not be closed, whereas other tags such as `<br>` and `<p>` can be closed during the pre-processing either to `<br/>`, `<p/>` or `<br> ...</br>` or `<p> ...</p>`. However, the presence of such tags in HTML invalidates the document for further processing by the XML Processor. To solve this problem, we used a text pre-processor written in Perl. This pre-processor tries to close the non-closed tags, if it is possible to close them, otherwise they are removed from the document since their removal will not affect the Arabic text mining. These cases reflect the unstructured-ness of HTML and make a case for XML as the candidate technology to be used in text-mining-specific applications.

After pre-processing all of the HTML documents, they are passed to the XML processor for further processing. The XML Processor uses rules written in XSL (Extensible Stylesheet Language) to convert HTML documents into XML. A sample XSL file includes the following specification:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html"/>
    <xsl:template match="/html"><xsl:apply-templates/></xsl:template>
    <xsl:template match="table/tr[position()=0]/td[position()=1]/font[position()=1]">
        <date> <xsl:apply-templates/> </date>
    </xsl:template>
    <xsl:template match="table/tr[position()=2]/td/font[position()=0]">
        <author> <xsl:apply-templates/> </author>
    </xsl:template>
    <xsl:template match="table/tr[position()=2]/td/font[position()=1]">
        <title> <xsl:apply-templates/> </title>
    </xsl:template>
    <xsl:template match="table/tr[position()=2]/td/font[position()=2]">
        <content><xsl:apply-templates/></content>
    </xsl:template>
    <xsl:template match="table/tr[position()=2]/td/font[position()=3]">
        <subject><xsl:apply-templates/></subject>
    </xsl:template>
</xsl:stylesheet>
```

This XSL specification uses templates to locate and extract the relevant information (author, data, title, content) and wrap them with the corresponding XML tags. The first template extracts the date at position (0, 1) of the table, enclosed within the second occurrence of `<font>` tags. Similarly, the name of the author, the title, the news content of the article, and the subject are located in position (2, 0), enclosed respectively within occurrence 0, 1, 2 and 3 of the tags `<font>` tags. We have used existing tools that provide parsing and translation facilities, such as Xerces from Apache software (Xerces), to implement the XML Processor. A sample of a transformed XML document has the following form:
4. TEXT PROCESSING

Figure 2 shows the stages of processing the Arabic documents.

After the text preprocessing stage every document is linguistically processed by the NLP model as follows:

4.1 Tokenization

Tokenization is a crucial stage in text processing. The tokenizer takes as input a plain text document and produces a set of tokens. Our tokenizer is fairly simple. It uses white space and punctuation marks as word separators. For example the Arabic sentence: “حليب الغنم أرخص من حليب البقير.” which means in English “Goats’ milk is cheaper than cows’ milk.” consists of 6 tokens: "حليب" between positions 0 and 4 in the text, "الغلام" between positions 5 and 10, “أرخص” between 11 and 15, “من” between 16 and 18, “حليب” between 19 and 21, “بقير” between 22 and 27, and lastly the token that contains the full stop “.” at position 28. Notice that there is no space between the last word in the query and the full stop. Full stops are used as one of the indicators of the end of the sentence.
4.2 The Morphological Analysis and Part-Of-Speech Tagging

Arabic is a derivational language in which morphology plays a significant role (Harmain 2001). Words in Arabic are classified into three types: nouns, verbs, and articles. Most Arabic morphemes are defined by three consonants, to which various affixes (prefixes, suffixes and infixes) can be added to create a word. For example, the tri-consonant "كتب" represents the concept of writing from which we can derive many words as follows:

- كتب → He wrote
- كاتب → a writer
- كتاب → a book
- كتب → books

There exists many more other derivations, but it is outside the scope of this paper to list them all. It is important to notice that all of these words have the letters "ك", "ت", and "ب" in common, and in the same order. Thus, the three consonants define the basic concept; while the affixes define the way that concept is applied. In actual Arabic, most of the affixes are applied in a regular and predictable way. However, there are some irregular words like "رجل" (man) which has the plural "رجال" (men). These irregular words have to be taken in account when implementing the morphological analyzer. After morphologically analyzing all tokens, this module attaches a part-of-speech tag to every token.

4.3 The Parser and Semantic Interpreter

Parsing is the process of analyzing language sentences using some grammar rules to find out their internal syntactic structure (Allen 1987). The Arabic sentence structure is different from that of English and other European languages. Arabic sentences can be classified as nominal جملة اسمية or verbal جملة فعلية. Nominal sentences begin with a noun such as: لولد ذكي (The boy is clever). Verbal sentences begin with a verb such as:

 Seamaniه ذهب الولد إلى المدرسة (The boy went to the school). Arabic linguists, since the days of Sebawi، have extensively studied the Arabic grammar. For parsing Arabic, all we need to do is to code what those linguists have specified. We don’t expect one to write grammar rules to parse every possible sentence in the language. However, the grammar rules have to be powerful enough to parse all basic sentence structure. Robust parsers are available for English (see for example BuChart in Gaizauskas, 1995) and it would be wise to try and port them to Arabic.

Semantic interpretation is the process of finding the meanings of sentences (Allen 1987). This process translates parse trees into a semantic structure. These are usually represented in explicit predicate-argument logical relations. Semantic interpretation can be carried out compositionally at the same time with parsing, or can be an independent stage (Gazdar 1989).

4.4 The Discourse Interpreter

Discourse interpretation is the most difficult stage of text processing (Allen 1987). The Discourse Interpreter aims to build an integrated model of the processed text by linking together the entities mentioned in the text. This model is then used by the results generator to produce the required results. In our case, the semantic net is used for text mining. One of the applications we are looking at is to find association rule across different documents. This is going to be our next goal.
5. CONCLUSION

We have defined the architecture of our Arabic Text Mining System and described how a natural language processing module can play a significant role in this system. We highly believe that the implementation of this architecture will open a big door of research in Arabic Text Mining and Arabic NLP. Both communities can join forces to achieve the ultimate goal of building robust Arabic text mining systems.

Our work is still at the early stages. A lot of work remains to be done. We can summarize our near future work as follows: Implement the Arabic text mining module; Implement a visualization tool for the results of the text mining system; and finally test and evaluate the system.

REFERENCES


