Automatic Translation of Arabic Text to Arabic Sign Language

Mohamed Mohandes
Electrical Engineering Dept, King Fahd University of Petroleum and Minerals
KFUPM 1885, Dhahran, 31261, Saudi Arabia
mohandes@kfupm.edu.sa

Abstract
Sign language is the basic means of communication among hearing-impaired people. Systems that could act as interpreters between vocal and hearing-impaired people would facilitate the life of deaf and integrate them in the society. Such systems should perform bidirectional translation of sign language and spoken language. A lot of efforts invested on translating sign languages to spoken languages. However, it is just as important to translate a spoken language to a sign language. This would provide a two-way communication between hearing-impaired and vocal people. A reasonably priced portable computer would be sufficient to perform the two-way translation. This paper proposes a system that translates Arabic text to Arabic sign language. A word that corresponds to a sign from the Arabic sign language dictionary calls a pre-recorded video clip showing the sign played on the monitor of the portable computer. If the word does not have a corresponding sign in the dictionary, it is finger spelled as done in real life by deaf for words that do not have specific signs; like proper names. They system is expandable as all what is needed is to add new signs to the database. The developed system can also be used to teach deaf or their relatives the Arabic Sign Language. It is accessible at the site http://naas.itc.kfupm.edu.sa:8080/SignsApp/.

Keywords: Arabic Sign Language, Text/speech translation to sign language, Gesture, Java programming, 1

1 Introduction
A sign language maps expressions of a certain language into a set of hand gestures enabling an individual to communicate by using gestures rather than by speaking. Sign Language forms the natural way of communication for deaf just like speech for vocal people. Hearing-impaired people often communicate with vocal people through a sign language interpreter. However, it is not always possible to find interpreters when needed; in addition to loosing privacy. Therefore, there is an increasing need to develop an automatic interpreter that is capable of translating signs to speech and the reverse using computers.

This becomes possible with the recent advancement of computers. It is now possible to have a hand held computer in an affordable price with enough processing power to perform this translation in real time. More efforts have been put to develop systems to recognize sign languages than systems to translate speech and text to sign languages. A complete translation system would interpret both: sign language to speech and speech to sign language. Unlike other sign languages, the Arabic sign language did not receive enough attention.

This paper describes a system to translate Arabic text to Arabic Sign language. This is a first step to translate speech to sign language with work also underway to translate vice versa, which is, sign language into speech. These two components may be combined to provide a two-way communication system between hearing-impaired and vocal people. The remaining of this section describes some of the work done on translating speech and text to other sign languages.

Cox et. al, [1] presented a system that translates the English speech to the British Sign Language (BSL) using a specially developed avatar. However, the system is constrained for post office operations. The system uses a phrase lookup approach due to the highly constraint environment in the post office. The authors divided the task to three different problems:

- Automatic speech to text conversion
- Automatic translation of arbitrary English text into suitable representation of sign language
- Display of this representation as a sequence of signs using computer graphics techniques.

The authors recognized that the simplest way of signing the set of phrases defined for the application is to store video-recordings of a person signing each phrase and concatenate the appropriate phrases in response to the output of the speech recognizer. However, they decided to use a virtual human (avatar) that they have already developed for other application. The developed system achieved accuracy of identification of the signed phrases of 61% for complete phrases and 81% for sign units. However, the feedback of deaf users and post office clerks were very encouraging for further development.
Suszczańska, et. al, [2] developed a system to translate texts written in Polish Language into the Polish Sign Language. They use Avatar as well with a dictionary of 600 signs.

Scarlatos, et. al, [3] introduced a system to translate speech into the American Sign Language (ASL). The system displays the ASL clips along with the written words. They used a built-in speech recognition engine in the Macintosh operating system. This added a limitation as this engine can only recognize words from a pre-defined set. They plan to extend the system to recognize more words and later for phrases.

San-Segundo, and others, [4] developed a system to translate speech into Spanish Sign Language. Their system is made up of four modules: speech recognizer, semantic analysis, gesture sequence generation and gesture playing. For the speech recognizer, they used modules developed by IBM. For the semantic analysis they used modules developed by the University of Colorado. For gesture sequence generation they use the semantic concepts associated to several Spanish Sign Language gestures. For gesture animation they developed an animated character and a strategy for reducing the effort in gesture generation. The strategy consists of making the system generate automatically all agent positions necessary for the gesture animation.

iCommunicator company has built a proprietary software [5] to translate English language to text or to the American Sign Language that costs several thousand dollars. A demo is provided in the web page of the company.

Eva Safar and Ian Marshall [6] described architecture of a system to translate English text into a variety of sign languages like Dutch, German, and British Sign Languages. They decomposed the process into two stages: manipulation of the English text into a semantic-based representation, and then the translation from this representation to graphically oriented representations which can drive a virtual avatar.

A group of 21 researchers at DePaul University [7,8] participated in developing an automated American Sign Language Synthesizer.

2 Description of the Developed System

Sign language maps letters, words or expressions of a certain language to a set of hand gestures thereby enabling an individual to communicate by using hands and gestures rather than by speaking. There has been a continuous increase in interest in facilitating communication with physically challenged people. Systems capable of recognizing sign-language symbols can be used as a means of one-way communication from deaf to vocal people. Many researchers are working along these lines including the Prince Salman Center for Disability Research. A recent project involves translating Arabic Sign Language to spoken words using an instrumented glove. However, this will only provide one way of communication. Much similar to how vocal people do not understand signs; deaf people cannot understand spoken words either. This is just one of the barriers preventing the full integration of deaf people into society.

The developed system is the first step towards the final goal of translating Arabic spoken language to sign language via voice recognition. This system may be displayed on a portable computer that acquires the speech and translates it immediately to sign language shown on the portable computer. The current stage of the project focuses on translating Arabic text to Arabic Sign Language.

When text is typed letter by letter, the system automatically starts searching for words that start with the letters being typed through an Arabic dictionary of words. Words that match with the letters that have been typed are displayed to the right. Upon entering a space character, the application recognizes the completion of a word and automatically displays the sign for the word best matched to what had been typed. In case of a mismatch, the letters of the words are finger spelled much similar to how people with hearing disability communicate.

An example of this process can be seen in the following figures. Upon entering a complete word, which isn’t in the database, images representing each letter of the word are displayed as can be seen in Figure 1. However, if the user were to have entered a word present in the video archive, such as in Figure 2, a video would have been displayed. Mismatched words with a large number of letters are displayed with small pictures so as to occupy less space as shown in Figure 3.

![Figure 1: The word that does not exist in the Dictionary, is finger-spelled](image1.png)

![Figure 2: Small word is displayed over the entire window](image2.png)

![Figure 3: The size of the letters of large words is displayed small to fit the window](image3.png)
3 Methods
In order to incorporate the scalability of web-based applications, this system was developed using Java Server Pages (JSP), implicitly having all the benefits of the Java programming language. For reasons of maintainability, user and cost friendliness, the Microsoft Access was chosen as the storage solution of choice. MS Access databases are portable, easy to access, use and deploy and were an ideal choice for our system. With functionality exposed via a typical web browser and implemented in state of the art JSP technology, this application is user-friendly, efficient in execution, and scalable.

3.1 SOFTWARE SPECIFICATION:

<table>
<thead>
<tr>
<th>Application</th>
<th>Web Application (Java Based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database:</td>
<td>MS Access</td>
</tr>
<tr>
<td>Technology:</td>
<td>Java, JSP, JavaScript, HTML</td>
</tr>
<tr>
<td>Web Server:</td>
<td>Apache Tomcat Server</td>
</tr>
<tr>
<td>Development Tool:</td>
<td>Net Beans IDE v4.1</td>
</tr>
<tr>
<td>Client-side Requirements:</td>
<td>Windows Media Player</td>
</tr>
<tr>
<td></td>
<td>ActiveX Control</td>
</tr>
</tbody>
</table>

3.2 DICTIONARY BASED
All entries of the database are brought into a dictionary object and are kept in its memory as shown in Figure 4. The dictionary contains Arabic words and the filenames that represent the visual sign of the associated word. Having the dictionary of words and associated filenames loaded in memory helps the application to search the given text very quickly. It minimizes the network traffic and unnecessary I/O hits with negligible memory costs. The matching will be very fast and unmatched entries will be responded to immediately. If the given word matches any entry in the dictionary then the application responds by first locating the file associated with the word being processed and then sending it to the user’s browser.

In conclusion, loading all entries of the database into memory allows faster access, better performance, reduced network traffic, and reduced database burden.

3.3 DATABASE
The application includes a database for holding all Arabic Signs of the Arabic sign language dictionary [9] along with associated file names used for matching words with their representative videos/images. Images and video files representing words in the Arabic dictionary are physical stored on the server at a special location. When the server receives a request for a particular sign, it will always look for the video under the aforementioned location. All the words visible in the word list of the application are also loaded from the database. This link with the database implies that manipulation of the word list is possible by merely adding a new row to the database table responsible for the word list. In fact, since the database is a MS Access database, a database that is inherently user friendly, adding, updating or deleting signs a relatively simple task in this application. Since the MS Access database is being cached in memory as a POJO (Plain Old Java Object), issues of scalability are dependent on the memory available to the server and not the database at hand.

At this point it should also be mentioned that the application is built in Java Technology using J2EE architecture and that to access the database it uses a standard JDBC-ODBC Bridge. So if needed, the MS Access database can be replaced by any ODBC-compliant, state of the art, high-scale database such as Oracle or DB2 as a future endeavor, without any major redevelopment.

As it is developed using Java it inherits all the advantages of using a state of the art, object oriented programming language. It can be deployed and setup on any Java compliant server such as Apache’s Tomcat, or IBM’s Websphere, etc. It can viewed by most browsers without any additional software. However, some browsers will require the Microsoft Windows Media Player ActiveX Control Plug-in to view the videos. Details on how to obtain this plug-in can be found directly on the main page of the application. The images of the sign alphabets are displayed directly on browser window with auto sizing facility with respect to the number of images presented.

3.4 HOW IT WORKS:
The index.html page has been divided into three frames as shown in Figure 5.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>top.jsp</td>
<td>Resides at the top, containing the top translate input box.</td>
</tr>
<tr>
<td>list.jsp</td>
<td>Resides on the right hand side displaying the video library entries.</td>
</tr>
<tr>
<td>display.jsp</td>
<td>Resides in the center. Responsible for showing the image(s) or the video.</td>
</tr>
</tbody>
</table>

The top.jsp fires events to the list.jsp and the display.jsp. The list.jsp fires events for the top.jsp only.
Event handling is a combination of client-side JavaScript and server-side Java Code translated to HTML, as shown in Figure 6.

This structure has the following advantages:

- The three files are separate and easy to manage
- Client-side JavaScript helps fast event handling for the application
- Since it's browser based, no client-side setting are needed to run the application except that the player and the code be available on the client machine
- Can be available on the web for any number of clients anywhere
- Powerful J2EE server-side coding ensures high availability and faster performance while employing fewer resources. (The total rendering time actually depends on the size of the video only and not on the server-side Java code execution.)

3.5 **JAVA PROGRAMMING LANGUAGE**

The application was developed using Java’s Object Oriented Programming Concept. Every thing is represented as an object. Some are serialized objects such that they can be persistent. The dictionary is loaded into the server memory on startup and is stored in a serialized object for quick matching of Arabic words. This application can be deployed on any Java web server and can be run from any Java enabled internet browser (IE, Netscape, Eudora, etc). This makes the application very user-friendly and flexible. To use the application the user can type in words or choose from the list of words shown. When the required word is complete, pressing Enter or Space will display of the corresponding sign. Figure 7 illustrates the actual flow of the program.
3.6 Video Editing/Formatting

A major concern was the widespread unavailability of high-speed internet, that is, appropriate bandwidth. The videos played for each word went through an intense process of optimization and compression to create file sizes that could be manageable over the internet available throughout the world.

Basic Dial-Up internet access at a rate of 56kbps was determined to be the baseline for usage of the web application. Over 1300 videos were optimized for playback on a dial-up 56Kbps connection under various conditions of bandwidth usage.

Each video file is now an optimum 25-30kb which takes less than 5 seconds to download on the baseline connection. Achieving this reduction in file size was done while keeping in mind the quality of the videos.

4 Conclusion

In this paper all words related to the signs of the Arabic Sign language Dictionary have been translated to signs. Specific efforts has been placed to reduce the size of the video clips so that the system is practical to be downloaded from a dial-up system in a reasonable time, while at the same time keeping good resolution of the clip to clearly show facial impression of the signs. The size of each video clips is reduced from around 500-600 KB to about 25-30 KB. The developed system constitutes the first phase of building a system that translates the Arabic speech to Arabic Sign Language.

Upon completion, the system can be integrated with the system that translates Arabic Sign Language to spoken Arabic Language to reach an integrated system that removes all barriers on the way of integrating deaf people with the rest of society. In the next phase, it is intended to obtain the root of every word related to a sign so that any of the derivatives of the word will play the same sign just like what deaf do on their daily life. Next, sophisticated system to translate speech to text will be integrated so that we obtain a system that translates speech to signs. The system is easily expandable as all what is needed is to add signs to the database. Therefore, during the search, the system will recognize the sign and call its video clip rather than fingerspell it. The developed system has been placed on the Internet and tested from several locations around the world.

5 Acknowledgements

The author would like to acknowledge the support of Prince Salman Centre for disability Research and King Fahd University of Petroleum and Minerals. Thanks also to Asad Nafees, Usman Ahmed, and Sadak Hussain for their help on the software development.

6 References


