Machine-Aided Spoken Language Evaluation: The Test Delivery Module

Brian Teaman, Osaka Jogakuin College
Steve McCarty, Osaka Jogakuin College
Takeshi Tamura, Kobe Institute of Computing

I. Introduction

In recent years, automatic speech recognition (ASR) has become common in many contexts. It is also slowly beginning to be found in software for computer assisted language learning. One recent application called *PhonePass* (2005) uses ASR to assess the speaking ability of non-native speakers of English using a normal telephone as the input device. The Machine-Aided Spoken Language Evaluation (MASLE) system seeks to go beyond the one-use *PhonePass* model by building a set of tools for the PC over the Internet so that many different types of speaking tests or interactions can be created so that speech can be rated by human or ASR programs.

The MASLE system (http://masle.org) is being developed with three different modules (shown in Figure 1, a test delivery system (1a-b), a rater jukebox (2a), human rater (2b) and an ASR component that rates the speech by computer (3).

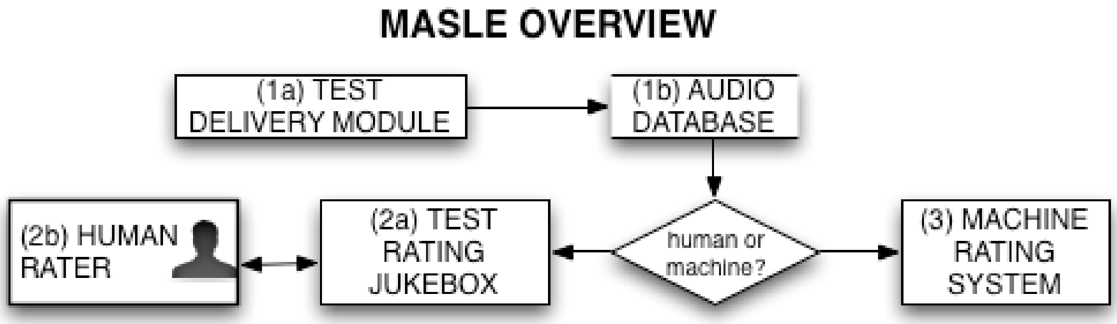


Figure 1: an overview of the MASLE system.

II. The Test-Delivery Module

This presentation will outline the test delivery module (Figure 1-1a-b) and demonstrate a working prototype. The goal of the test delivery module is to provide a flexible interface for delivering prompts and collecting audio recordings of the test-takers' spoken language on a stand-alone computer, LAN or the Internet. The system will provide prompts to a test-taker and then record the language that is spoken by her. In order to perform these tasks there are four parts needed, shown in Figure 2: 1. A program that will run over the web, 2. recording software, 3. an input database, 4. an output database and finally 5. the test-taker.

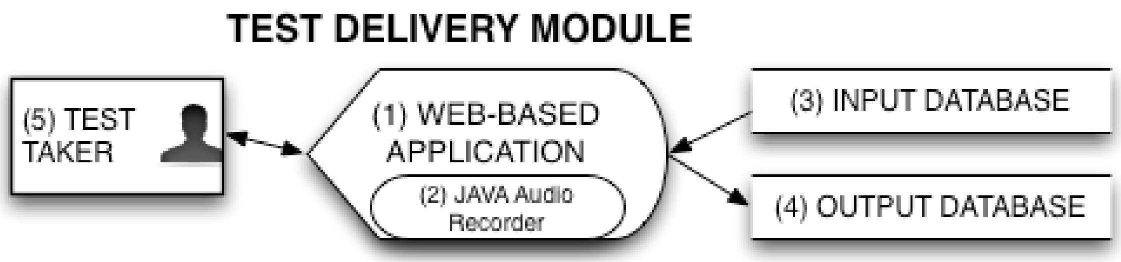
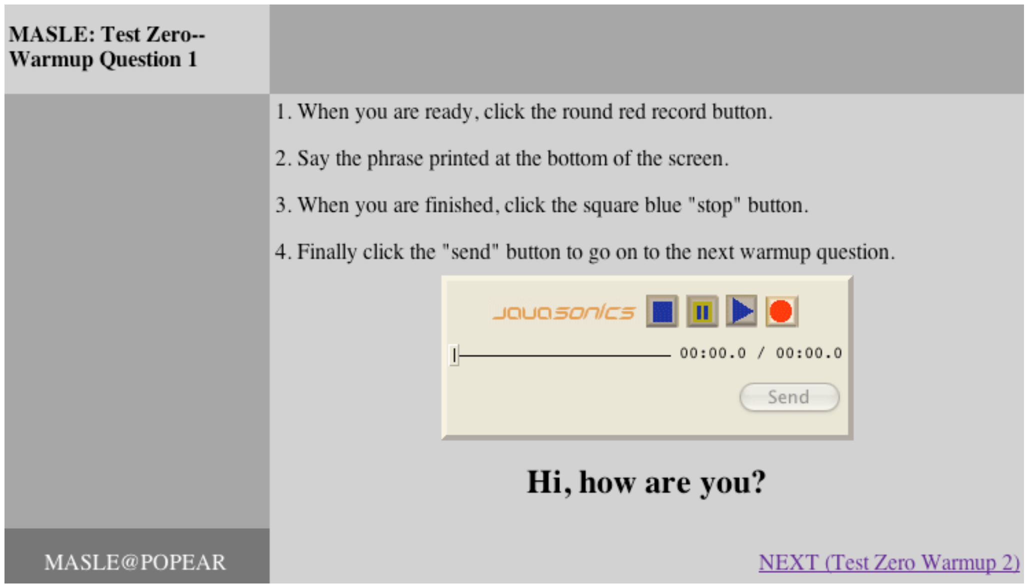


Figure 2: The structure of the test delivery module.

First, is the core program or web-based application (Figure 2-1). It is written in XHTML with PHP (v. 4.3) as the server control language. The program will run through a web browser and access the input database (Figure 2-3) in order to create the appropriate display and then record the speech using the embedded recording program (Figure 2-2). Figure 3 shows a snapshot of one screen from the web-based application including the embedded recorder. When this screen is displayed, there is pre-recorded audio file which will read the instructions out loud to the user.



MASLE: Test Zero--
Warmup Question 1

1. When you are ready, click the round red record button.

2. Say the phrase printed at the bottom of the screen.

3. When you are finished, click the square blue "stop" button.

4. Finally click the "send" button to go on to the next warmup question.

Hi, how are you?

MASLE@POPEAR

NEXT (Test Zero Warmup 2)

Although the instructions in this example are simultaneous audio and text, it could easily be in only audio or only text depending on the goals of the test. In the upper left-hand corner is the information about the test name and item number. The main body of the screen shows the instructions for what the test taker is to do. The recorder, shown as the lighter portion of the main box in figure 3 is the JAVA based recorder called ListenUp (2005). It is necessary for the user to control the java recorder manually with four actions. These four actions are:

1. Clicking the record button (the red dot).
2. Speaking as directed. In this case, saying: "Hi, How are You?"
3. Clicking the stop button (the square button).
4. Uploading the recorded audio to the server by clicking the ***send*** button.

This recorder can be programmed in various ways. For example, rather than using the stop button, it can be programmed to stop after some preset length of time. It is also possible to change the arrangement or shape of the buttons. The recorder also needs to be told in which format to record in order to allow for different sampling rates and audio compression rates so these features are also programmable. The task that test-takers need to perform on this question item is to read a phrase. However, it is possible to use audio, video or images as prompts. For example a brief video or animation could be played and the speaking task would be to describe what the test-taker saw happen. Finally, the speaker clicks the link at the bottom right to go on to the next page.

The input database mentioned in figure 2-3 will contain the following data:

- Display text, including test prompts as well as other meta instructions to direct the user in how to the test.
- Supplemental media of any sort needed for the test including audio, graphics or video.
- The name and location of the audio file to be saved.

With this database, the program will be instructed what to display as a prompt for each item (whether it be text, audio, some visual object or some combination of the three). It will also know what to name the audio file that will be recorded for the output database. Using PHP, it is possible to simply have just one PHP driven test-item webpage and then use this database to specify what to display each time this same page is called. The test or interaction then would be totally data-driven, so that the program would display different prompts from the database depending on the task. PHP combined with a database therefore provides a powerful tool for creating flexible tests and interactions that can be changed by simply creating a new database.

The output database (Figure 2-4) contains the following information:

- The speaker ID.
- Test ID and date.
- The item ID.
- The audio data.

The speaker ID, test ID and date will be handled by the name of the folder. Therefore the folder named ***809.0.05.10.23*** could easily be understood by human or computer as indicating:

- The speaker ID is *809*.
- The test ID is *0*.
- The date is October 23, 2005.

The file names within the folder ***809.0.05.10.23*** are then given unique names such as ***01.wav**, **02. wav***, ***03.wav*** and so on. With this file structure, test items can be accessed by the human or machine rating modules (Figure 1-2 and 1-3).

A session then runs as described in the following. After some preliminary screens displaying instructions and familiarizing the learner with the test, the browser will display the first test prompt while instructing the user to perform the task so the users' speech can be recorded and uploaded to the Internet. After the recorded item is sent to a server and stored, the program goes on to the next test item where more speech will be recorded and uploaded. This repeats as many times as it is necessary until the test is finished.

III. Conclusion

This prototype of the test-delivery system represents the first step in the three major parts of the MASLE project. It shows that a reasonably flexible testing system can be constructed for language testing over the Internet. The PHP language with some version of XHTML can be combined with a java-based audio recorder and a database to become a powerful team that enables this type of testing. Currently, there are limitations with the interface that require the user to do too much of the work. With time, tweaking of this interface and advances in networked capabilities of computer systems, this will undoubtedly improve. For example, it would be nice to be able start the recording automatically after some pre-determined period of time. Burke (2005) reports that although this is not currently a built-in feature, Javascript could be used to accomplish this. If the audio were started and stopped automatically, it would eliminate some of the potential errors that could be introduced by the test-taker actually controlling the recorder. However, in the current state it represents a solid first step in seeing the entire MASLE system become a reality.

References

Burke, Phil (2005) Personal communication by e-mail with author of Listenup on September .

ListenUp version 1.54 (2005) Computer program. Softsynth.com.

PhonePass (2005). A phone-based test of spoken English. Ordinate Corporation.

Teaman, Brian (2004). An Introduction to MASLE: Machine Aided Spoken Language Evaluation. Hiroshima University Journal of Language Teaching and Research, Vol. 7:39-49.

Acknowledgement:

This research was partly supported by the Ministry of Education, Science and Culture of Japan under Grants-in-Aid No. 17520406.