An Undergraduate Course on Speech Recognition Based on the CSLU Toolkit

Ben Serridge

Universidad de las Américas, Puebla, México

ABSTRACT

This paper describes an undergraduate course in speech recognition, based on the CSLU Toolkit, which was taught at the Universidad de las Américas in Puebla, México. Throughout the course, laboratory assignments based on the toolkit guided students through the process of creating a recognizer, while in-class lectures consistently refereed to the architecture of the toolkit as a concrete example of an existing system.

The class was organized so that lectures and laboratory assignments followed the steps taken in the creation of a new recognizer. The students first recorded and labeled their own corpus, then proceeded to design and train neural network based recognizers, before finally testing for performance and creating sample applications. As a final project, students performed simple, well-defined experiments using the recognizers they had constructed.

The CSLU Toolkit is freely available for non-commercial use from http://cslu.cse.ogi.edu/. In future, similar courses based on the toolkit could be created and shared by many researchers in the speech community via the world-wide web.

1. INTRODUCTION

As with any other technical subject, a course in speech recognition is much enhanced by laboratory assignments, through which students gain hands-on experience with the technology and the satisfaction of creating systems of their own. In many cases, however, the effort required to transform existing research systems into laboratory assignments suitable for students is prohibitive. The CSLU Toolkit [1,2] provides a framework which facilitates the creation of new laboratory assignments and which provides students with an easy-to-use environment within which they can learn about phonetic theory, train their own recognizers, and develop sample applications.

The organization of this paper follows the organization of the course, in which students created their own domain-specific recognizers from scratch using the CSLU Toolkit.

2. RECORDING A CORPUS

The first step in creating a recognizer for a new domain is to record a corpus. In this particular course, the domain was that of continuously spelled words, and the first laboratory assignment was for each student to record his or her own voice, as well as the voices of two friends. The recording sessions were controlled by a dialogue created previously using the Rapid Application Developer (RAD), a graphical user interface to the CSLU Toolkit. Figure 1 shows the dialogue used for the recordings as viewed from within RAD.

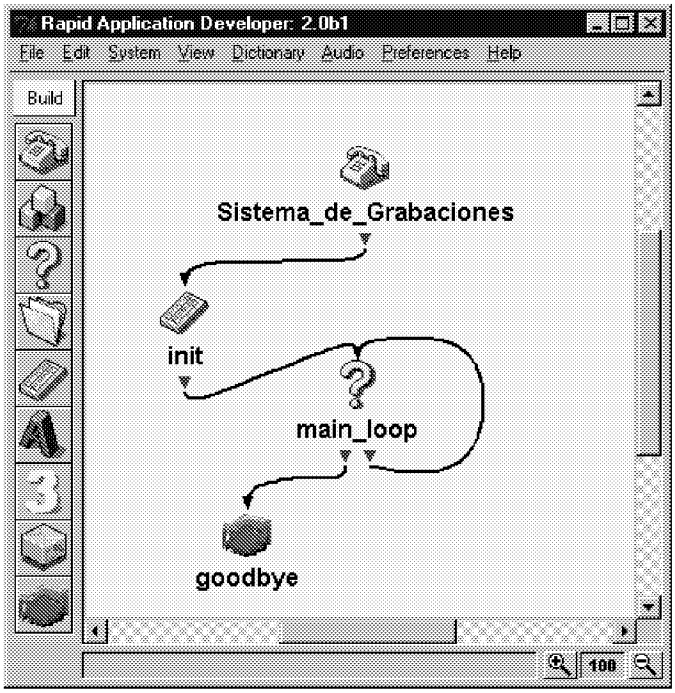

Figure 1: The graphical interface of the Rapid Application Developer, a part of the CSLU Toolkit. The telephone icon establishes a connection to the input device (either a microphone or the telephone) and the keyboard icon does some initial pre-processing and plays a welcome prompt. The main part of the application is the recognition loop in which each prompt is given and the resulting speech, instead of being recognized, is stored to a file in the corpus. The specifics of the operations associated with each icon are defined by Tcl code accessible through the RAD interface.

The corpus of 50 speakers was divided (randomly) into training, development, and test sets of 30, 10, and 10 speakers, respectively. Each speaker recorded 20 spelled words, divided as follows:

- Words 1-5 were responses to questions such as "Please spell your first name."
- Words 6-10 were nonsense words with a high frequency of rare letters such as X, Q, and K.

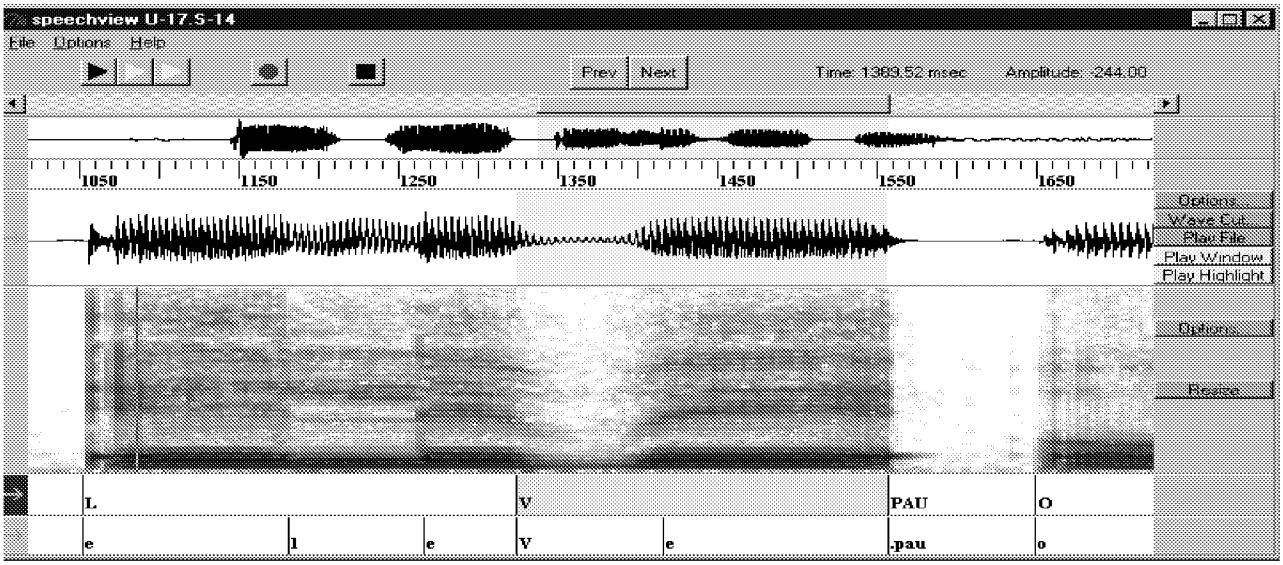

Figure 2: An example screen of Speech Viewer. The first set of labels is at the word level, and the word "V" has been highlighted. (The fact that words, in this domain, are in fact letters of the alphabet, forces students to understand the difference between letters and phonemes, which in Spanish are more often than not represented by the same symbol.) The second set of labels are the phones.

- Words 11-20 were real words randomly chosen from an electronic Spanish dictionary.

Recordings were made using a head-mounted microphone, but could also have been made via telephone.

3. LABELING A CORPUS

The next step in creating a new recognizer is to label, at the phonetic level, the data in the training set. Although in many cases this labeling can be done automatically using forced-alignment, for educational purposes it is very useful for students to go through the process by hand. Although the task is arduous, it gives students an appreciation for allophonic variation and coarticulatory effects. The task of labeling complements the phonetic theory described in class and gives a first hand understanding of the difficulties involved in recognizing fluent speech.

The labeling task is accomplished by an application called "Speech Viewer," shown in Figure 2, which displays a waveform, its spectrogram, and any labels associated with it. Boundaries between labels can be adjusted using the mouse, and labels can also be inserted or deleted.

To make the labeling task easier, students first labeled each speaker at the word level. Rather than fill in word-level labels from scratch, artificial labels were first generated automatically (since we know what each speaker was supposed to have said) using a heuristic algorithm that predicts the duration of each word based on average phone durations. Once the word-level boundaries had been adjusted, phonetic labels were then generated using the same algorithm and similarly adjusted by hand. The task of placing boundaries between adjacent phones forced students to confront the disparity between phonetic theory and reality, while at the same time giving them additional experience with the spectral representation of speech. Each student was responsible for the labeling of two speakers, a task which takes about four hours.

The Speech Viewer tool is also useful for laboratories designed to exhibit specific speech properties, as it allows the user to listen to the part of the speech signal associated with any label. For example, one could design a special "corpus" through which students must search in order to find examples of particular effects such as glottalization, coarticulation, nasalization, etc.

4. TRAINING A RECOGNIZER

Before training on the labeled data, several design decisions must be made regarding the units to be recognized. First, the user must specify whether to divide each phonetic unit into one, two, or three parts. The user must also define contextual classes, which allow data from similar contexts to be shared in training each unit. Both decisions require a consideration of the tradeoffs involved between the number of phonetic units to be trained and the amount of data available to train each unit. In the case of the letters domain, only a subset of the phones are used, and many of these only within very narrow contexts. However, the small size of the corpus meant that the sparse data problem could not be ignored altogether.

The first laboratory assignment involved the design of the recognizer, in which students must define the number of parts to use for each phone as well as the contextual classes. The following assignment involved the actual training. By collapsing several steps of the training procedure into a few easy-to-use training scripts, the training process was greatly simplified, at the expense of freedom regarding some design parameters. (For example, students could not choose to use a different set of measurements, nor could they choose to train an

HMM-based recognizer, although the toolkit provides facilities for doing so.)

5. FINAL PROJECTS AND SAMPLE APPLICATIONS

As a final project, each student was asked to design and execute a well-defined experiment measuring the effects of changing a single parameter. Experiments included training gender-dependent recognizers, training recognizers trained on data down-sampled to 8 kHz, and measuring the effects of changes on the number of parts for each phone or the number of contextual classes. These experiments were fairly easy to implement and required almost no writing of code.

Finally, students were able, using RAD, to use their own recognizers in sample applications. While the final projects had focused on experimental results in which the speech is rarely, if ever, actually heard by the user, the sample applications gave a more concrete example of what a real system might be like.

6. CONCLUSIONS

This paper described how the CSLU Toolkit can be used as a basis for laboratory assignments in a course in speech recognition. The Tcl interface to existing libraries and the Rapid Application Developer make the professor's task of creating laboratory assignments straightforward, while the graphical interfaces to Speech Viewer and RAD are easy for students to use. Furthermore, the toolkit can be used consistently as an example implementation of the concepts learned in class. Finally, the Toolkit, including synthesis, recognition, corpus-labeling, and application development, is freely available. In future, similar courses based on the toolkit could be created and shared by many researchers in the speech community via the world-wide web.

7. ACKNOWLEDGEMENTS

The structure of the course, including the use of final projects and the letters domain, is based on the Automatic Speech Recognition class (6.345) taught by Victor Zue and Jim Glass at MIT. In that class, the laboratories are based on SUMMIT, a segment-based speech recognition system developed by the MIT Spoken Language Systems group.

8. REFERENCES

1. Sutton, S., Novick, D. G., Cole, R., and Fanty, M., "Building 10,000 spoken-dialogue systems." Proceedings of the International Conference on Spoken Language Processing, Philadelphia, PA, Oct., 1996.

2. Schalkwyk, J., de Villiers, J., van Vuuren, S., and Vermeulen, P., "CSLUsh: An Extendible Research Environment", EUROSPEECH, 1997.