Interactive ASR Error Correction for Touchscreen Devices

David Huggins-Daines
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dhuggins@cs.cmu.edu

Alexander I. Rudnicky
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
air@cs.cmu.edu

Abstract

We will demonstrate a novel graphical interface for correcting search errors in the output of a speech recognizer. This interface allows the user to visualize the word lattice by "pulling apart" regions of the hypothesis to reveal a cloud of words simlar to the "tag clouds" popular in many Web applications. This interface is potentially useful for dictation on portable touchscreen devices such as the Nokia N800 and other mobile Internet devices.

1 Introduction

For most people, dictating continuous speech is considerably faster than entering text using a keyboard or other manual input device. This is particularly true on mobile devices which typically have no hardware keyboard whatsoever, a 12-digit keypad, or at best a miniaturized keyboard unsuitable for touch typing.

However, the effective speed of text input using speech is significantly reduced by the fact that even the best speech recognition systems make errors. After accounting for error correction, the effective number of words per minute attainable with speech recognition drops to within the range attainable by an average typist (Moore, 2004). Moreover, on a mobile phone with predictive text entry, it has been shown that isolated word dictation is actually slower than using a 12-digit keypad for typing SMS messages (Karpov et al., 2006).

2 Description

It has been shown that multimodal error correction methods are much more effective than using speech alone (Lewis, 1999). Mobile devices are increasingly being equipped with touchscreens which lend themselves to gesture-based interaction methods.

Therefore, we propose an interactive method of visualizing and browsing the word lattice using gestures in order to correct speech recognition errors. The user is presented with the decoding result in a large font, either in a window on the desktop, or in a full-screen presentation on a touchscreen device. If the utterance is too long to fit on the screen, the user can scroll left and right using touch gestures. The initial interface is shown in Figure 1.

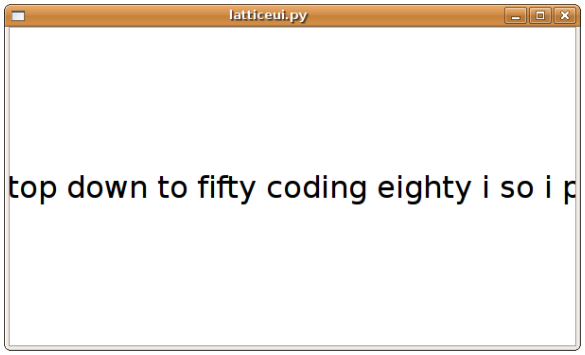

Figure 1: Initial hypothesis view

Where there is an error, the user can "pull apart" the result using a touch stroke (or a multitouch gesture where supported), revealing a "cloud" of hypothesis words at that point in the utterance, as shown in Figure 2.

It is also possible to expand the time interval over which the cloud is calculated by dragging sideways, resulting in a view like that in Figure 3. The user can then select zero or more words to add to the hypothesis string in place of the errorful text which was "exploded", as shown in Figure 4.

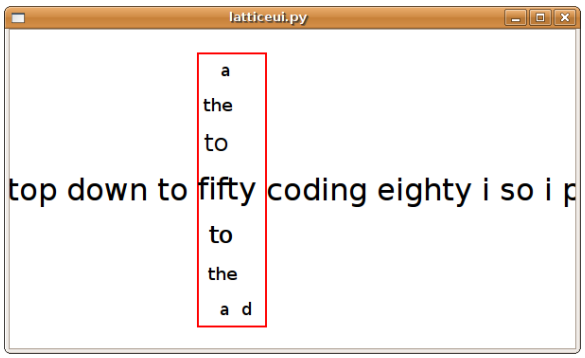

Figure 2: Expanded word view

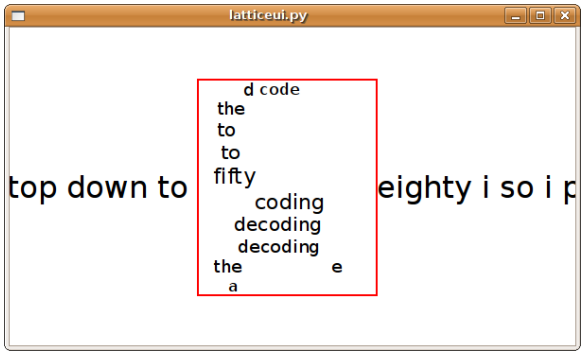

Figure 3: Word cloud expanded in time

The word cloud is constructed by finding all words active within a time interval whose log posterior probability falls within range of the most probable word. Word posterior probabilities are calculated using the forward-backward algorithm described in (Wessel et al., 1998). Specifically, given a word lattice in the form of a directed acyclic graph, whose nodes represent unique starting points t in time, and whose edges represent the acoustic likelihoods of word hypotheses w_s^t spanning a given time interval (s, t), we can calculate the forward variable $\alpha_t(w)$, which represents the joint probability of all word sequences ending in w_s^t and the acoustic observations up to time t, as:

$$\alpha_t(w) = P(O_1^s, w_s^t) = \sum_{v_s^t \in prev(w)} P(w|v)P(w_s^t)\alpha_s(v)$$

Here, $P(w|v)$ is the bigram probability of (v, w) obtained from the language model and $P(w_s^t)$ is the acoustic likelihood of the word model w given the observed speech from time s to t, as approximated by the Viterbi path score.

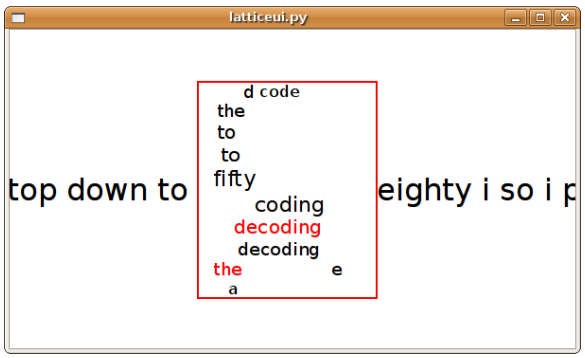

Figure 4: Selecting replacement words

Likewise, we can compute the backward variable $\beta_t(w)$, which represents the conditional probability of all word sequences beginning in w_s^t and the acoustic observations from time $t+1$ to the end of the utterance, given w_s^t:

$$\beta_t(w) = P(O_t^T|w_s^t) = \sum_{v_t^e \in succ(w)} P(v|w)P(v_t^e)\beta_e(v)$$

The posterior probability $P(w_s^t|O_1^T)$ can then be obtained by multiplication and normalization:

$$P(w_s^t|O_1^T) = \frac{P(w_s^t, O_1^T)}{P(O_1^T)} = \frac{\alpha_t(w)\beta_t(w)}{P(O_1^T)}$$

This algorithm has a straightforward extension to trigram language models which has been omitted here for simplicity.

This interface is inspired by the web browser zooming interface used on the Apple iPhone (Apple, Inc., 2008), as well as the Speech Dasher lattice correction tool (Vertanen, 2004). We feel that it is potentially useful not only for automatic speech recognition, but also for machine translation and any other situation in which a lattice representation of a possibly errorful hypothesis is available. A video of this interface in Ogg Theora format[1] can be viewed at http://www.cs.cmu.edu/~dhuggins/touchcorrect.ogg.

[1]For Mac OS X: http://xiph.org/quicktime/download.html
For Windows: http://www.illiminable.com/ogg/downloads.html

3 Script Outline

For our demonstration, we will have available a poster describing the interaction method being demonstrated. We will begin by describing the motivation for this work, followed by a "silent" demo of the correction method itself, using pre-recorded audio. We will then demonstrate live speech input and correction using our own voices. The audience will then be invited to test the interaction method on a touchscreen device (either a handheld computer or a tablet PC).

4 Requirements

To present this demo, we will be bringing two Nokia Internet Tablets as well as a laptop and possibly a Tablet PC. We have no requirements from the conference organizers aside from a suitable number of power outlets, a table, and a poster board.

Acknowledgements

We wish to thank Nokia for donating an N800 Internet Tablet used to develop this software.

References

E. Karpov, I. Kiss, J. Leppänen, J. Olsen, D. Oria, S. Sivadas and J. Tian 2006. Short Message System dictation on Series 60 mobile phones. *Workshop on Speech in Mobile and Pervasive Environments (SiMPE) in Conjunction with MobileHCI 2006.* Helsinki, Finland.

Keith Vertanen 2004. *Efficient Computer Interfaces Using Continuous Gestures, Language Models, and Speech.* M.Phil Thesis, University of Cambridge, Cambridge, UK.

Apple, Inc. 2008. *iPhone: Zooming In to Enlarge Part of a Webpage.* http://docs.info.apple.com/article.html?artnum=305899

Roger K. Moore 2004. Modelling Data Entry Rates for ASR and Alternative Input Methods. *Proceedings of Interspeech 2004.* Jeju, Korea.

James R. Lewis 1999. Effect of Error Correction Strategy on Speech Dictation Throughput *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting.*

Frank Wessel, Klaus Macherey, Ralf Schlüter 1998. Using Word Probabilities as Confidence Measures. *Proceedings of ICASSP 1998.*