Parakeet: A Demonstration of Speech Recognition on a Mobile Touch-Screen Device

Keith Vertanen and Per Ola Kristensson
Cavendish Laboratory, University of Cambridge
JJ Thomson Avenue, Cambridge UK
{kv227,pok21}@cam.ac.uk

ABSTRACT
We demonstrate Parakeet – a continuous speech recognition system for mobile touch-screen devices. Parakeet's interface is designed to make correcting errors easy on a handheld device while on the move. Users correct errors using a touch-screen to either select alternative words from a word confusion network or by typing on a predictive software keyboard. Our interface design was guided by computational experiments. We conducted a user study to validate our design. We found novices entered text at 18 WPM while seated indoors and 13 WPM while walking outdoors.

Author Keywords
Mobile continuous speech recognition, touch-screen interface, error correction, speech input, word confusion network

ACM Classification Keywords
H.5.2 User Interfaces: Voice I/O

INTRODUCTION
This is a demonstration companion paper to [2]. In this paper, we describe our work on a system called Parakeet. Parakeet allows users to dictate text while on the move. Our system consists of a speech recognition engine (based on PocketSphinx [1]) and a novel interface for performing corrections. Parakeet is designed to make mobile continuous speech recognition pleasant and efficient.

INTERFACE DESCRIPTION
Parakeet runs on mobile Linux devices, such as the Nokia N800 (figure 1). To enter text, users speak into a wireless microphone. While the user is speaking, audio is streamed to a continuous speech recognizer which is running on the actual device. Once recognition is complete, the result is displayed in the form of a word confusion network (figure 2).

The best recognition hypothesis is shown at the top. Each column contains likely alternatives for each recognized word. At the bottom, a series of delete buttons allow words to be removed. The user can scroll left or right by touching the arrow buttons on either side of the screen. Users make corrections by using a number of different actions:

Figure 1. The Parakeet system running on a Nokia N800 device.

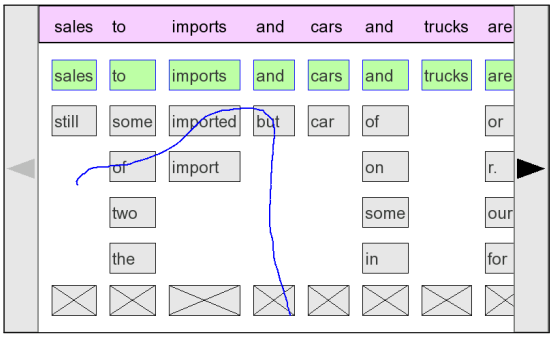

Figure 2. Parakeet's main correction interface. The recognition result is shown at the top. Likely alternative words are displayed in each column. In this example, the user is changing several words and deleting another word in a single crossing action.

- **Tapping** – An alternate word can be chosen by simply tapping on it. The selected word is displayed in green.
- **Crossing** – Multiple words can be corrected in a single continuous crossing gesture (figure 2).
- **Copying** – Words can be dragged between different columns or inserted between columns (figure 3).
- **Replacing with variant** – By double-tapping a word, a morphological variant can be chosen (figure 4).
- **Typing** – Arbitrary corrections can be made using a predictive software keyboard (figure 5). As a user types, word completion predictions are offered.

IUI'09, February 8-11, 2009, Sanibel Island, Florida, USA.
ACM 978-1-60558-331-0/09/02.

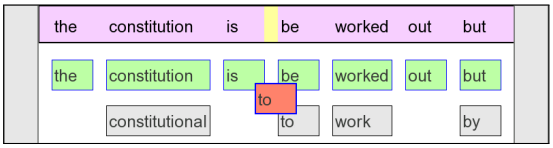

Figure 3. The user is inserting the word "to" between "is" and "be" by dragging it from its original column to the desired location.

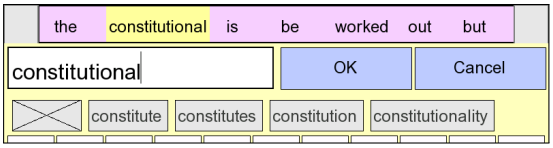

Figure 4. After touching the word "constitutional", the user is brought to the predictive software keyboard. The morphological variants for "constitutional" are shown in a row above the keyboard.

Our design was guided by computational experiments on recorded audio. Figure 6 shows how different design choices affected error correction efficacy. Among other things, increasing the number of word alternatives allowed more successful corrections. However, the majority of the gains were observed using a small number of words in each column (we chose to display five). We found that by always providing a delete button, successful corrections improved substantially. Copying words between columns and replacing words with their morphological variants provided further gains.

USER STUDY

To see how well our system worked in practice, we conducted an initial user study with four participants. The aim was primarily to validate our design. We had participants speak and then correct newswire sentences while seated indoor and while walking outdoors. Our main findings were:

- **Error rates** – Users experienced a word error rate (WER) of 16% indoors and 25% outdoors. After correction by the user, the WER was 1% indoors and 2% outdoors.
- **Entry rates** – Users' average text entry rate was 18 words per minute (WPM) indoors and 13 WPM outdoors. This included somewhat long recognition delays (average 22s). Without these delays, entry rates would have increased by almost a factor of two.

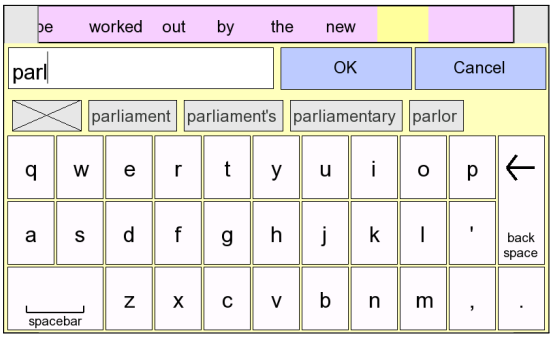

Figure 5. The predictive software keyboard. The user has typed "parl". The most likely word completions are displayed above the keyboard.

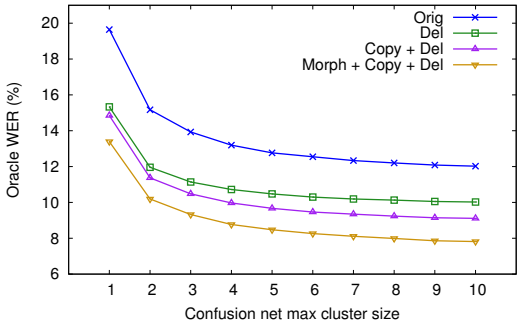

Figure 6. Oracle word error rate (WER) as a function of cluster size in the confusion network. The top line is using the original confusion network with no modifications. The other lines show how error rate decreased as we added more correction features.

- **Use of confusion net** – Users performed corrections via the confusion net interface when possible. We found that when errors could be completely corrected using the confusion net, users did so 96% of the time.
- **Touch versus crossing** – 90% of selections used a touch action, 10% used a crossing action. Crossing actions were particularly popular for selecting delete boxes.
- **Copying** – Copying a word between columns was not a popular feature and was only used 3 times.
- **Predictive keyboard** – When users typed a word on the keyboard, they used word completion 54% of the time. On average, users typed 3 letters before selecting a prediction.

CONCLUSIONS

Parakeet is a system for mobile text entry using speech recognition. In our user study, we found novices could effectively use Parakeet both indoors while seated and outdoors while walking. We plan on improving our system based on our initial user study before performing a large-scale evaluation.

ACKNOWLEDGMENTS

We thank Nokia for partially funding our research and donating the N800. The following applies to P.O.K. only: The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 220793.

REFERENCES

1. D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proc. of the IEEE Conf. on Acoustics, Speech, and Signal Processing*, pages 185–188, May 2006.
2. K. Vertanen and P. O. Kristensson. Parakeet: A continuous speech recognition system for mobile touch-screen devices. In *IUI '09: Proc. of the 14th Intl. Conf. on Intelligent User Interfaces*. ACM, 2009.