SPEECHALATOR: TWO-WAY SPEECH-TO-SPEECH TRANSLATION IN YOUR HAND

Alex Waibel[1&4], *Ahmed Badran*[1], *Alan W Black*[1&2], *Robert Frederking*[1], *Donna Gates*[1]
Alon Lavie[1], *Lori Levin*[1], *Kevin Lenzo*[2], *Laura Mayfield Tomokiyo*[2]
Juergen Reichert[4], *Tanja Schultz*[1], *Dorcas Wallace*[1], *Monika Woszczyna*[3], *Jing Zhang*[4]
[1] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA
[2] Cepstral, LLC, [3] Multimodal Technologies Inc, [4] Mobile Technologies Inc.
speechalator@speechinfo.org

ABSTRACT

This demonstration involves two-way automatic speech-to-speech translation on a consumer off-the-shelf PDA. This work was done as part of the DARPA-funded Babylon project, investigating better speech-to-speech translation systems for communication in the field. The development of the Speechalator software-based translation system required addressing a number of hard issues, including a new language for the team (Egyptian Arabic), close integration on a small device, computational efficiency on a limited platform, and scalable coverage for the domain.

1. BACKGROUND

The Speechalator was developed in part as the next generation of automatic voice translation systems. The Phrasalator is a one-way device that can recognize a set of pre-defined phrases and play a recorded translation, [1]. This device can be ported easily to new languages, requiring only a hand translation of the phrases and a set of recorded sentences. However, such a system severely limits communication as the translation is one way, thus reducing one party's responses to simple pointing and perhaps yes and no.

The Babylon project addresses the issues of two-way communication where either party can use the device for conversation. A number of different groups throughout the US were asked to address specific aspects of the task, such as different languages, translation techniques and platform specifications. The Pittsburgh group was presented with three challenges. First, we were to work with Arabic, a language with which the group had little experience, to test our capabilities in moving to new languages quickly. Second, we were instructed to use an interlingua approach to translation, where the source language is translated into an intermediate form that is shared between all languages. This step streamlines expansion to new languages, and CMU has a long history in working with interlingua based translation systems. Third, we were constrained to one portable PDA-class device to host the entire two-way system: two recognizers, two translation engines, and two synthesizers.

2. RECOGNITION

We used an HMM-based recognizer, developed by Multimodal Technologies Inc, which has been specifically tuned for PDAs. The recognizer allows a grammar to be tightly coupled with the recognizer, which offers important efficiencies considering the limited computational power of the device. With only minor modification we were able to generate our interlingua interchange format (IF) representation directly as output from the recognizer, removing one module from the process.

MTI's recognizer requires under 1M of memory with acoustic models of around 3M per language. Special optimizations deal with the slow processor and ensure low use of memory during decoding. The Arabic models were bootstrapped from the GlobalPhone [2] Arabic collections as well as data collected as part of this project.

3. TRANSLATION

As part of this work we investigated two different techniques for translation, both interlingua based. The first was purely knowledge-based, following our previous work [3]. The engine developed for this was too large to run on the device, although we were able to run the generation part offline seamlessly connected by a wireless link from the handheld device. The second technique we investigated used a statistical training method to build a model to translate structured interlingua IF to text in the target language. Because this approach was developed with the handheld in mind, it is efficient enough to run directly on the device, and is used in this demo.

4. SYNTHESIS

The synthesis engine is Cepstral's Theta system. As the Speechalator runs on very small hardware devices (at least small compared to standard desktops), it was important that the synthesis footprint remained as small as possible.

The speechalator is to be used for people with little exposure to synthetic speech, and the output quality must be

very high. Cepstral's unit selection voices, tailored to the domain, meet the requirements for both quality and size. Normal unit selection voices may take hundreds of megabytes, but the 11KHz voices developed by Cepstral were around 9 megabytes each.

5. ARABIC

The Arabic language poses a number of challenges for any speech translation system. The first problem is the wide range of dialects of the language. Just as Jamaican and Glaswegian speakers may find it difficult to understand each other's dialect of English, Arabic speakers of different dialects may find it impossible to communicate.

Modern Standard Arabic (MSA) is well-defined and widely understood by educated speakers across the Arab world. MSA is principally a written language and not a spoken language, however. Our interest was in dealing with a normal spoken dialect, and we chose Egyptian Arabic; speakers of that dialect were readily accessible to us, and media influences have made it perhaps the most broadly understood of the regional dialects.

Another feature of Arabic is that the written form, except in specific rare cases, does not include vowels. For speech recognition and synthesis, this makes pronunciations hard. Solutions have been tested for recognition where the vowels are not explicitly modeled, but implicitly modeled by context. This would not work well for synthesis; we have defined an internal romanization, based on the CallHome [4] romanization, from which full phonetic forms can easily be derived. This romanization is suitable for both recognizer and synthesis systems, and can easily be transformed into the Arabic script for display.

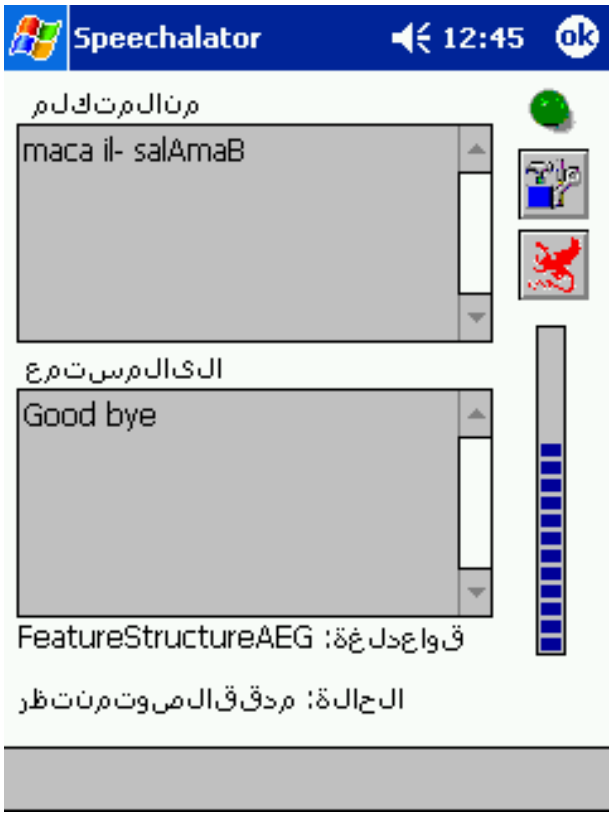
6. SYSTEM

The end-to-end system runs on a standard Pocket PC device. We have tested it on a number of different machines, including various HP (Compaq) iPaq machines (38xx 39xx) and Dell Axims. It can run on 32M machines, but runs best on a 64M machine with about 40M made available for program space. Time from the end of spoken input to start of translated speech is around 2-4 seconds depending on the length of the sentence and the actual processor. We have found StrongARM 206MHz processors, found on the older Pocket PCs, slightly faster than XScale 400MHz, though no optimization for the newer processors has been attempted.

Upon startup, the user is presented with the screen as shown in Figure 1. A push-to-talk button is used and the speaker speaks in his language. The recognized utterance is first displayed, with the translation following, and the utterance is then spoken in the target language. Buttons are provided for replaying the output and for switching the input to the other language.

7. DISCUSSION

The current demonstration is designed for the medical interview domain, with the doctor speaking English and the patient speaking Arabic. At this point in the project no formal evaluation has taken place. However, informally, in office-like acoustic environments, accuracy within domain is well over 80%.



Arabic input Screen
Speechalator snapshot

8. REFERENCES

[1] Sarich, A., "Phraselator, one-way speech translation system," http://www.sarich.com/translator/, 2001.

[2] T. Schultz and A. Waibel, "The globalphone project: Multilingual lvcsr with janus-3," in *Multilingual Information Retrieval Dialogs: 2nd SQEL Workshop*, Plzen, Czech Republic, 1997, pp. 20–27.

[3] A. Lavie, et al. "A multi-perspective evaluation of the NESPOLE! speech-to-speech translation system," in *Proceedings of ACL 2002 workshop on Speech-to-speech Translation: Algorithms and Systems*, Philadelphia, PA., 2002.

[4] Linguistic Data Consortium, "Callhome egyptian arabic speech," 1997.