Two-Way Speech-to-Speech Translation on Handheld Devices

Bowen Zhou, Daniel Déchelotte and Yuqing Gao

IBM T. J. Watson Research Center
Yorktown Heights, New York 10598
{zhou, ddechel, yuqing}@us.ibm.com

Abstract

This paper presents a two-way speech translation system that is completely hosted on an off-the-shelf handheld device. Specifically, this end-to-end system includes an HMM-based large vocabulary continuous speech recognizer (LVCSR) for both English and Chinese using statistical n-grams, a two-way translation system between English and Chinese, and, a multilingual speech synthesis system that outputs speech in the target language. This paper describes the system development and the functionality of the major components, focusing on the optimization efforts employed to achieve real time results on such a limited platform.

1. Introduction

In recent years, there have been significant efforts to develop reliable and satisfactory automatic speech-to-speech translation systems, which are typically available on powerful resources such as desktop servers or laptop computers. However, it is noted that such devices are not compact, and thus are not convenient for mobile applications. This limits the usefulness of such translation technologies. Many realistic circumstances can only be effectively aided by truly mobile devices such as Personal Digital Assistants (PDA). Furthermore, to overcome the requirements of wireless connection and other limitations, the entire end-to-end system should be hosted on the PDA [1].

Automatic speech-to-speech translation is a highly complex task. Moreover, a large amount of computation is involved to achieve reliable translation performance. Resources are not just computation limited, but memory and storage requirements, and the audio input and output requirements all tax current systems to their limits. In order to bridge the gap between contemporary translation systems and the current mobile computing platforms, we have employed a number of optimizations, significantly enhancing the accessibility of our automatic speech translation technology. Recently, we presented a speech-to-speech translation system transforming spoken English to Mandarin Chinese running on a PDA [1], demonstrating the feasibility of such ideas. In this paper, we further extend our previous work, and present a complete two-way speech-to-speech translation system that is deployed on an off-the-shelf PDA on an embedded Linux platform. The presented system employs the same architecture as its desktop counterpart, the MASTOR [2, 3] system. Particularly, the system maintains a LVCSR that operates in real time, or near-real time.

2. Background and System Overview

2.1. System Architecture

Our previous work in speech-to-speech translation systems [2, 3] included the development of a complete system running on a desktop or a laptop computer. We have maintained the basic architecture of this earlier system, which can be found in [1, 2], while adapting it to the capabilities of a handheld computer.

2.2. Hardware and Software Specification

To demonstrate the feasibility of building a system with our speech translation architecture on a standard PDA, we select the target hardware platform as HP (né Compaq) iPaq PDA model H3800, a popular and widely available device. The processor is Intel's StrongARM CPU running at a frequency of 206 MHz. The system has 64 MB RAM and 32 MB flash ROM memory. Excluding the memory required by the operating system (OS), the available SDRAM memory for running our end-to-end speech translation system is about 35 MB. The original iPaq is shipped with Microsoft's Pocket PC 2002 as the OS, In our work, the OS is replaced with Familiar (version 0.6.1)[4], a full featured Linux distribution for the iPaq series, based on the embedded Linux kernel. For audio I/O, this iPaq uses a Philips audio codec UDA1341, and provides a push-and-talk button and a built-in microphone for speech input, as well as an integrated speaker for audio output.

The PDA is further expanded during development using an IBM CompactFlash Microdrive (or a MultiMediaCard) to provide additional storage, mainly for the n-gram language models and acoustic models, statistical models for parsing and generation, as well as bilingual dictionaries.

To sufficiently make use of existing system components found in the MASTOR desktop system, all of the major components are ported to this PDA platform and a number of portability issues have to be addressed. In addition, numerous optimizations are required to make the applications fit in handheld devices' tiny physical memory. To construct applications for such an ARM-Linux based handheld device, a cross-compiler tool chain is employed running on a Linux based x86 host.

3. System Development

3.1. LVCSR on a PDA

The recognition module developed for our mobile speech translation system is an HMM-based LVCSR engine using statistical n-grams. Unlike most grammar-based embedded speech recognition systems, our system has the advantages of large vocabulary coverage and flexibility to switch to new application

This work is founded by the DARPA BABYLON project.

domains, which are typically only found on desktop-based systems. To accomplish this, IBM's large vocabulary speech recognition system, as featured in the popular ViaVoice recognizer, was ported to the ARM processor architecture.

3.1.1. *Porting LVCSR to Handheld Devices*

The LVCSR system takes as input speech sampled at a rate of 22 KHz. The acoustic front-end uses a 24-dimensional cepstral feature vector extracted every 10 ms. Blocks of 150 ms features are transformed using linear discriminant analysis (LDA) into a 40-dimensional feature vector.

For our two-way speech translation task, we need to build both English and Mandarin Chinese speech recognition systems. The English recognizer uses an alphabet of 52 phones. Each phone is modeled with a 3-state left-to-right hidden Markov model (HMM). This system has approximately $3,500$ context-dependent states modeled using more than $40,000$ Gaussian distributions. The context-dependent states are generated using a decision-tree classifier. Similarly, the Chinese recognizer employs an identical HMM structure as English, using an alphabet of 162 phones. However, the acoustic feature set is different since Chinese is a tone-dependent language, and thus the pitch-contour is included among the features.

On porting this large scale system to ARM architecture, it is first noted that the StrongARM platform (as well as most currently available handheld devices), unlike the Intel x86 series, has no integrated floating point (FP) hardware. It depends entirely on software that emulates the FP co-processor. Despite much of the IBM recognizer being developed to use mostly integer computations, our initial profiling experiments showed that substantial amounts of time were consumed by FP calculations. Therefore, significant efforts were made to integerize most of the signal processing front-end and search components of this system. This includes a fixed point math implementation of the following major recognizer components: the Mel-cepstrum feature extraction, the Gaussian likelihood computation of the context dependent phone models, as well as the procedures of fast match and detailed match during the decoding process. Particularly, at the feature extraction front end, all major computation modules such as high-pass filtering, discrete cosine transformation, LDA, pitch calculation and silence detection have been mostly integerized. In addition, although not a major portion of the compute time, some gains were achieved by making use of Intel's Integrated Performance Primitives library which is optimized for the ARM architecture, for the fast Fourier transform calculations used prior to Mel-band energy calculation.

Although all of the computationally expensive portions of the code have been integerized, floating point (FP) calculations are still used in some small portions of the recognition engine. To avoid the expensive overhead of kernel traps for FP calculation, software implementations are employed to emulate FP calculations as library calls in our ported system. To enable this, a customized tool chain is locally built in this work, where the libraries in this tool chain are completely recompiled to support the "soft float" scheme.

Statistical tri-gram language models have been used for this continuous speech recognizer as ported to the iPaq platform. For our specified application domain of force protection and medical triage (FPMT), the English system has a vocabulary of 10K words, and the Chinese system has a vocabulary size of 8K. This recognizer also includes all of the support of the custom acoustic enrollments as featured in the ViaVoice recognizer. Use of these enhancements is part of our future work. Currently, support for interactive enrollment on the iPaq has not been ported, so another desktop computer must be used to complete the model building for a specific speaker. However, the recent addition of non-interactive training should make implementation of speaker dependent adaptation much more straightforward.

Currently, the continuous speech recognition system typically functions in real time for English, and a little over real time for Chinese due to the extra overhead of pitch calculation. However, decoding utterances in mismatched acoustic environments or out of domain can significantly increase the computation time.

3.1.2. *Running Two LVCSR Engines on a PDA*

One of the major challenges in developing this two-way speech translation system is that large vocabulary speech recognition needs to be arranged on this limited platform for two distinct languages. This further complicates the porting efforts in maintaining satisfactory recognition accuracy and speed for both languages with memory constraints.

To reduce memory footprint, two recognition systems can be activated in turn according to the current direction of translation. However, switching recognition systems on the fly will typically introduce undesirable computational overhead and time delay. To avoid this, both the English and Chinese recognition in our system are activated in memory through multiple connections to a recognition engine via IBM SMAPI (speech manager applications programming interface). Since these two systems employ distinct acoustic models (AM) and language models (LM), little memory (only the binary libraries) can be shared between the two processes. Thus, the images of AM and LM have to be mapped to virtual memory. In total, running two recognition processes in parallel requires about 12 MB physical memory.

Another issue is raised by integerized calculation of the engine. Fixed point code necessarily involves scaling to effectively use the available dynamic scope of the processor word size. In most cases, the amount of shifting was empirically determined using reference recordings. It is noted that Chinese and English speech recognition systems employ different acoustic feature sets and thus have different dynamic ranges. Consequently, a particular scaling used in integerized calculation that is effective for one language may provide poor resolution or overflow for another language. Therefore, the amount of shifting needs to trade off between two languages using reference recordings. As a consequence, the embedded speech recognition system will not have the same flexibility in effectively dealing with signals of unusually high or low volume.

3.1.3. *Compensating the Effects of the iPaq Microphone*

It should be noted that the integrated microphone on PDAs are substantially different from the high-quality headphone microphones used in our desktop systems. In addition, the microphone is not a "close-talk" one and therefore has characteristics known to degrade speech recognition. To compensate the effects of this low-quality microphone, significant efforts have been made to improve the quality of acoustic models used for the iPaq. A large amount of speech data in both English and Chinese has been collected using multiple iPaq devices (includ-

Table 1: A comparison of acoustic models (WER%)

Language AM	English	Chinese
Baseline	23.77	25.38
Adapted	17.37	13.96

Table 2: Compare integerized and regular FP engine (WER%)

Task	English		Chinese	
Sys.	M	F	M	F
Regular FP	19.88	15.89	12.43	13.17
Integerized	25.36	19.36	17.36	16.15

ing both H3600 and H3800 with the assumption that their microphones share similar acoustic characteristics). A simple program was designed to allow user to dictate to the iPaq following given text prompts, which are generally in the domain of FPMT.

The speech is sampled at 22050 Hz and stored on an IBM microdrive (the microdrive noise is intentionally captured). The recording are typically performed in regular office environments with insignificant background noise (such as keyboard strokes, footsteps, etc). Excluding some significantly corrupted audio segments, we totally collected speech from 191 unique English speakers and 96 unique Chinese speakers, which represent about 13.7 hours of Chinese speech with more than 13K utterances, and about 20 hours of English speech with more than 22K utterances. Based on the collected data, supervised MAP-MLLR adaptation have been performed to achieve more robust gender-dependent models for the iPaq, deriving from general acoustic models used in ViaVoice dictation products.

3.1.4. Recognition Experimental Results

This ported recognizer for handheld devices is evaluated on an H3800 iPaq using the heldout test data collected on the iPaq. For English, we reserved 160 utterances from 16, gender balanced, speakers for testing. The Chinese test data is composed of 160 utterances collected from 15 speakers (7 females and 8 males). The experiments are all conducted using gender-dependent acoustic models.

The first set of experiments are designed to measure the effectiveness of adapted acoustic models. Identical LMs are used for each language, which are interpolated using general dictation and in domain LMs. These experiments have been performed on desktop computers and the regular floating point recognizer is used. The reported results are averaged from male and female experiments. Compared with the baseline, as indicated in Table 1, adapted models significantly improves both Chinese and English recognition. Table 2 depicts the accuracy comparison between the ported integerized engine and the regular floating point engine. These results indicate that despite the heavy optimizations for fixed point calculations to make the engine usable on the iPaq platform, the performance degradation compared with the FP engine is within expectation. It is also noted that the degradation varies with different tasks, showing that the empirically determined scaling scheme works better for some tasks than for others.

3.2. Translation Module on a PDA

As described in [1], the translation module is composed of a statistical natural language understanding (NLU) and a statistical natural language generation (NLG) module. The entire translation procedure of this system takes less than real time for our domain.

The NLU module is based on the statistical parser employed in IBM telephony natural language dialog systems [5]. This component utilizes statistical decision-tree models to determine the meaning and structure of the input utterance, which is achieved by assigning a hierarchical tree structure to the recognized sentence as predicted by the statistical model. While the NLU module is not a significant computational bottleneck, it is important to improve the runtime speed of this module to lower the overall response time of the system. An effort was made in this work to reduce the runtime memory requirements and to improve the parsing speed. Primarily, this involved a different implementation of parsing the xml-based model structures and parameters. As a result, both memory overhead and the time of module initialization are significantly reduced. In addition, the “soft-float” emulation libraries, as described in Sec. 3.1, were used as well to improve the speed of FP calculation. Due to the dynamic range of the probabilities in the NLU models, it is unlikely that an integerized version of this search would work as well. For this two way translation system, two sets of decision tree based models have been built for parsing Chinese and English spoken sentences respectively. The total runtime memory requirement is about 12 MB. Presently, the parsing speed is about 1.4 second per utterance for our current application domain. Detailed in domain performance running on an iPaq can be found from Table 3 in [1].

Next, high level semantic translation is performed by a natural language generation system based on the semantic representation obtained from NLU module. The statistical natural language generation algorithm is introduced in our previous study based on maximum entropy modeling [2]. For Chinese generation, the methods discussed in [6] have also been reimplemented and merged. Language model based rescoring is also applied in English generation. On porting this component, this NLG module has been reimplemented to fit with low computational resources available on a PDA. This includes a more efficient implementation of search procedures, as well as significantly reduced I/O routines. The generation module itself and associated models are compact on iPaq platform, which takes about 4 MB runtime memory for both languages. In order to decrease the memory requirement of the bilingual dictionary from the one used in our desktop system, a set of new dictionaries have been created for our domain with reduced size. About 9K entries are included for English to Chinese translation, and about 15K entries are used for the reverse direction.

3.2.1. Translation Experiments

The translation module on the iPaq is evaluated using automatically transcribed audio data. A total of 237 spoken English utterances from 3 speakers and 77 spoken Chinese utterances from 2 speakers are used as the test data. The audio was extracted from role-playing human conversations using close-talk microphones recorded on desktop computers. Automatic transcripts (with an average WER around 10% for English and about 5% for Chinese) generated by a regular recognizer running on desktop are fed to the translation module of both desk-

Table 3: Two-way translation results (BLEU scores)

system direction	Desktop	iPaq
English-Chinese	0.3517	0.3069
Chinese-English	0.2866	0.2795

top and iPaq systems. Table 3 summarizes the translation performance as BLEU [7] scores, where only one reference is used in the evaluation. It is clear that the iPaq translation module maintains comparable performance with its desktop counterpart, while the performance loss is mainly caused by the effects of reduced bilingual dictionaries.

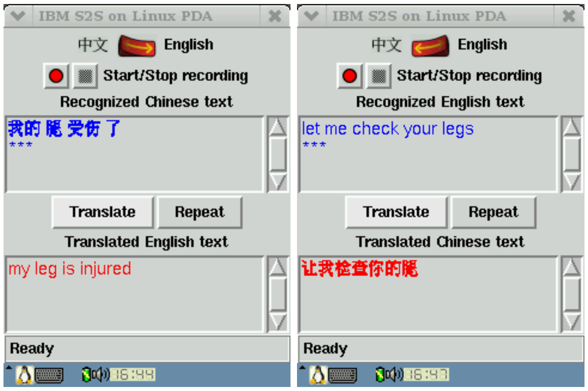
3.3. Text-to-Speech (TTS) Synthesizer

The utterance in target language generated from the translation module is synthesized by a text-to-speech engine. Considering the limited resources available for a mobile device, the current TTS system is based on IBM's multi-lingual formant TTS technology. The formant based TTS system can synthesize an unlimited number of voices, and allows flexible customizations by modifying vocal characteristics such as gender, pitch, volume and speed. It also has the advantage of supporting unlimited vocabularies. In effect, the TTS software can pronounce any text that it is given. More importantly, format-based TTS system has a small memory footprint, which makes it appropriate to be deployed in embedded applications.

To make the IBM TTS engine run efficiently on a StrongARM architecture, the computations are heavily integerized. In addition, a certain portion of the code base has been rewritten in this work to increase the portability of the engine, particularly, to address a number of portability issues that are raised by the peculiarities of the target architecture including the OS, the compiler, and the CPU characteristics. For example, pointer alignment and memory overlay issues used to make the TTS system unstable for Chinese TTS, where multiple bytes are used to represent a single character. After addressing these issues, as well as performing other performance optimizations in this work, the TTS engine runs stably for both Chinese and English on this StrongARM Linux platform. During runtime, the TTS component takes a total of 5 MB memory for both Chinese and English synthesis.

3.4. System Integration and User Interface

Figure 1 shows two screenshots of this two-way system running on iPaq Linux. The top line of the GUI indicates the current direction of speech translation with an arrow button. As we maintain major system components residing in memory, the user can seamlessly switch the translation direction by clicking this arrow button. The system also uses a status bar, which locates at the bottom of the GUI, providing the user with feedback about the system status. Compared to our previous version described in [1], the users now are provided with two ways to activate a translation task. The users can either press and hold the push-and-talk button and release the button after speech input, or click the record and stop menu-bar on the screen to start and terminate recognition.



我的 腿 受伤 了
***

my leg is injured

Ready

let me check your legs
***

让我检查你的腿

Ready

Figure 1: Screenshots of IBM Two-way Speech-to-Speech Translation System Running on a Linux-based iPaq

4. SUMMARY

This paper presented a two-way speech-to-speech translation system, which is completely hosted on an off-the-shelf PDA running embedded Linux. This mobile translation system distinctively includes an HMM-based large vocabulary continuous speech recognizer using tri-gram language models. Moreover, this PDA-based system maintains comparable translation accuracy found in its desktop counterpart.

5. ACKNOWLEDGMENTS

The authors thank Fu-hua Liu and Ruhi Sarikaya for adapting the acoustic and language models for our application domain. Also thank Jeffrey Sorensen, Lubos Ures, Charles Prosser, Tony Lee, Jerry Quinn for their help with this work.

6. References

[1] B. Zhou, Y. Gao, J. Sorensen, D. Déchelotte, and M. Picheny, "A Hand-held speech-to-speech translation system," in *Proc. IEEE ASRU 2003*, Dec. 2003.

[2] B. Zhou, Y. Gao, J. Sorensen, Z. Diao, and M. Picheny, "Statistical natural language generation for speech-to-speech machine translation systems," in *Proc. ICSLP 2002*, Sept. 2002.

[3] Y. Gao, B. Zhou, Z. Diao,J. Sorensen,and M. Picheny, "Mars: A statistical semantic parsing and generation based multilingual automatic translation system," *Machine Translation*, 17(3), pp. 185 - 212, 2002

[4] "http://familiar.handhelds.org," website.

[5] K. Davies et al., "The IBM conversational telephony system for financial applications," in *Eurospeech*, 1999

[6] L. Gu, Y. Gao, and M. Picheny, "Improving statistical natural concept generation in interlingua-based speech-to-speech translation," in *Eurospeech-2003*, Sept. 2003.

[7] K. Papineni,S. Roukos,T. Ward,and W. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *ACL*, 2002.