Real-Time Speech-to-Speech Translation for PDAs

R. Prasad, K. Krstovski, F. Choi, S. Saleem, P. Natarajan, M. Decerbo, D. Stallard

BBN Technologies
50 Moulton Street, Cambridge, MA 02138, USA
{rprasad,pnataraj}@bbn.com

Abstract - **In this paper we present a speech-to-speech translation system configured for translingual communication in English and colloquial Iraqi on a mobile, handheld device. The end-to-end system employs a medium/large vocabulary n-gram speech recognition engine for recognizing English and colloquial Iraqi, a question canonicalizer for mapping a recognized English question or command to one of the questions supported in the system, a concept translation engine for translating recognized Iraqi text, and a text-to-speech synthesis engine for playing back the English translation for the Iraqi to the English speaker. In addition to describing the system architecture and the functionality of the components, we present optimization techniques that enable low-latency, real-time speech recognition on low-power hardware platforms.**

I. INTRODUCTION

English-speaking field personnel in foreign lands often need to communicate with residents of the host country who do not speak English. In a crisis situation, there is little time to train these personnel in the host country language, and human interpreters will often be in short supply. Portable devices for speech-to-speech (S2S) language translation would therefore be very useful in such environments. These devices will also have a far reaching impact in the commercial sector, in applications such as portable language translation for travelers.

Under the DARPA TransTac and Babylon programs, various teams including BBN have developed systems that enable two-way communication over a language barrier [1-3]. Most of these systems adopt either a “2-way” [2-3] approach or a “1.5-way” [1] approach. The 2-way systems seek, in principle, to translate any utterance, by using general statistical models trained on large amounts of speech and text data. The 1.5-way systems use a task-directed approach to make the problem easier by specifying a fixed set of English questions with pre-recorded foreign language translations, and a fixed set of Arabic answers/concepts that can be translated into English.

The 1.5-way’s advantages and disadvantages, both stem from its constrained dialog schema. A key advantage of 1.5-way systems is clarity: unlike the 2-way approach, the user always knows what the system said to the foreign-language respondent, and the respondent always hears a fluent and intelligible recorded translation which increases the likelihood of the desired response.

BBN has previously developed a 1.5-way S2S system for medical/refugee processing domain [1]. The prototype in [1] was configured for common-off-the-shelf (COTS) laptop and the target language was Levantine Arabic. More recently, to overcome the shortcomings of the 2-way and 1.5-way approaches, BBN has developed a robust S2S solution that is a novel synthesis of 1.5-way and 2-way approaches [4]. The new, improved prototype facilitates bi-directional information exchange in English-Iraqi. It employs BBN’s state-of-the-art large vocabulary speech recognition engine for English and Iraqi speech recognition, for translation it couples BBN’s concept translation engine with BBN’s statistical machine translation (SMT) engine, and for text-to-speech (TTS) synthesis it uses Iraqi and English TTS from Cepstral LLC. The handheld-based S2S translation system described in this paper is a limited version of our laptop prototype in [4].

The key challenge here, as in [2-3], is to perform medium-to-large vocabulary automatic speech recognition (ASR) and machine translation in a computationally efficient manner so as to enable conversation at a normal pace while running on resource-limited hardware.

This paper is structured as follows. In Section II, we describe the system architecture for integrating the various component technologies and the user interface for the end-to-end translation system. Section III describes optimization techniques used for developing a small footprint version of BBN’s ASR engine. In Section IV, we describe the English ASR configuration used in the end-to-end system. In Section V, we discuss the details of Iraqi ASR configuration and our solution to mitigating the large vocabulary problem posed by the inflective nature of Iraqi. In Section VI, we describe the translation engines used in the S2S system. In Section VII, we present our conclusions and directions for future work.

II. SYSTEM ARCHITECTURE AND USER INTERFACE

Figure 1 illustrates the concept of operations for the 1.5-way approach that we have adopted for limited hardware platforms. As shown in Figure 1, the English ASR engine generates a text output for the spoken utterance from the user. The English text is then processed by a canonicalizer which maps the recognized text to one of the questions supported. A pre-recorded foreign language audio recording corresponding

1-4244-1039-8/07/$25.00 ©2007 IEEE

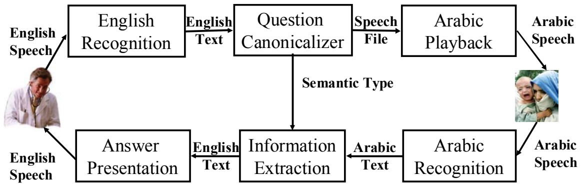


Figure 1: Concept of operations for BBN's Speech-to-Speech translation system for PDAs.

to the English question is then played out to the foreign-language respondent.

Although the set of supported questions (typically in the hundreds) is fixed, with the combination of *n*-gram ASR and the canonicalizer allows for different ways of asking the same question. The foreign-language speaker responds to the played-out question in his/her native language. The foreign-language ASR engine generates a text output for the respondent's speech. The text output is then sent to the concept translation engine. The concept translation [1] engine uses the semantic type of the question to generate an English translation of the concepts identified in the foreign-language response. Finally, the English text is played back to the English speaker using a TTS engine.

Figure 2 illustrates the architecture we have developed for integrating the components required for the end-to-end speech-to-speech (S2S) translation system. At the center of the operations is the application manager within the S2S Graphical User Interface (GUI). Except for the English TTS system, which is dynamically linked with the S2S GUI, the other four components are encapsulated into separate processes. These processes communicate with the GUI via Windows system messages.

Windows CE imposes a 32 MB virtual memory size limit per process. This severe memory restriction ruled out incorporating all four components of the S2S system into a single process. By separating each component into its own process, we were able to provide a separate 32 MB virtual memory per component. Since Iraqi Arabic is a large vocabulary problem, we implemented a novel memory management code that allows the Arabic ASR engine to use more than 32 MB of virtual memory.

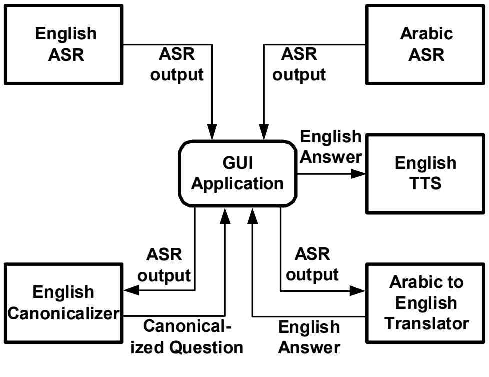
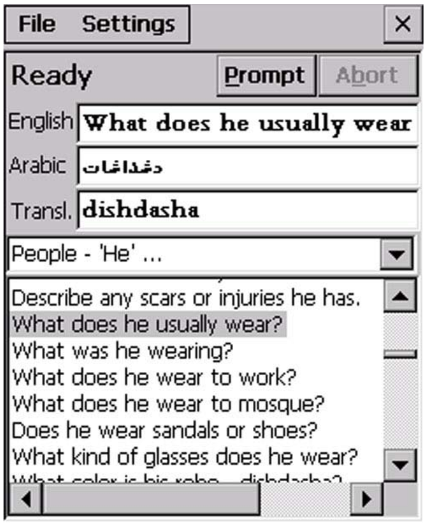


Figure 2: System architecture for Speech-to-Speech translation on handheld devices.



Ready

English What does he usually wear

Arabic دشداشات

Transl. dishdasha

People - 'He' ...

Describe any scars or injuries he has.
What does he usually wear?
What was he wearing?
What does he wear to work?
What does he wear to mosque?
Does he wear sandals or shoes?
What kind of glasses does he wear?
What color is his robe – dishdasha?

Figure 3: User interface for BBN's Speech-to-Speech translation system on handheld devices.

The screenshot of the GUI for the end-to-end S2S system is shown in Figure 3. Hardware buttons available on the handheld computers are used to trigger events for recognizing English and Arabic. The English ASR result is displayed in the text box labeled "English" and the Arabic ASR results is displayed in the text box labeled "Arabic". Following canonicalization, the English recognition result in the "English" text box is replaced by the canonical form. The English translation for the Arabic speech recognition output is displayed in the text box labeled as "Transl.". In addition to the speech input, the system provides the capability for the English speaker to play the desired question or command by double-clicking on the question displayed in the lower half of the interface. The questions are organized into categories, and the category to be displayed can be selected through the pull-down menu. The GUI also provides software buttons for playing the question again as well as aborting the play-back of a question.

The end-to-end S2S application is available on the Phraselator 2 (P2) and HP iPAQ handheld computers. The P2 handheld employs a 433 MHz Intel PXA255 processor and the HP iPAQ we used is equipped with a 624 MHz Intel PXA270 processor. Although slower than the iPAQ, the P2 has 256 MB of user accessible RAM memory compared to the 128 MB available on the HP iPAQ. The P2 runs the standard platform of the Windows CE operating system and the HP iPAQ runs the Pocket PC 2003 platform.

III. SMALL FOOTPRINT ASR

A key challenge in developing an end-to-end bi-directional S2S translation system for handheld computers is to incorporate two ASR engines on the limited hardware. In this section, we describe the different techniques we have used to develop a small footprint version of the BBN Byblos^{TM} ASR engine.

Integerization of the BBN Byblos^{TM} ASR Engine: The Byblos ASR engine uses phonetic hidden Markov models (HMM) with one or more forms of the following parameter tying: Phonetic-Tied Mixture (PTM), State-Tied Mixture (STM), and State-Clustered-Tied Mixture (SCTM) models. The states of each phonetic model are clustered based on the triphone or quinphone context into different "codebooks" (groups of Gaussian components). The mixture weights are clustered using the linguistically-guided decision trees.

The acoustic front-end in Byblos extracts a 45-dimensional feature vector every 25 milliseconds. Decoding is performed in two passes [5]. The forward pass uses PTM or STM acoustic models and a composite set bigram language model (LM). The output of the forward decoding pass consists of the most likely words at each frame, along with their partial forward likelihood scores. The backward decoding pass operates on the output of the forward pass using SCTM acoustic models and an approximate trigram LM to either generate a 1-best hypothesis, or an N-Best list.

The StrongARM processor available on handheld computers is an integer processor. Floating point operations can only be performed via software emulation and, as a result, are prohibitively slow for ASR. Therefore, we integerized the feature extractor as well as the search engine to run efficiently on the StrongARM platform. During search, to mitigate the loss in precision due to use of integer computations, we compute the scores in the log-domain. We used pre-computed tables for computing logarithms. Also, log-adds, i.e. computing *log(a+b)* when *log(a)* and *log(b)* are known were performed efficiently using pre-computed values for *log(1+ (exp(log(b) – log(a))*.

The integerized version of the Byblos decoder resulted in a negligible loss in accuracy over the floating-point version.

Compact Acoustic Models: Due to limited resources available on handheld computers, we use acoustic models with significantly less parameters than the ones used for the laptop systems. Specifically, on the laptop we use detailed SCTM models, whereas the ASR engines on the handheld computer are configured with smaller PTM models.

Caching Gaussian Distances for Multi-pass Recognition: As mentioned earlier in the section, the BBN Byblos decoder performs a two pass recognition search. On the laptop the acoustic models used in the two passes are usually different. For ASR on handheld, unlike the laptop we use the same acoustic model in both decoding passes. Doing so enables us to reuse the Gaussian distances computed in the forward pass for the backward pass. This is achieved by caching the Gaussian distances for each frame in an array.

Fast Gaussian Computation: Typically, computing the acoustic score for each active theory is the most compute intensive step in the search for the most likely word sequence. To speed-up the Gaussian distance computation, we use Gaussian shortlists [6]. We also quantized the means and variances of the Gaussians in our acoustic model to reduce the memory and computational requirements.

Entropy Pruning of N-grams: In addition to using compact acoustic models, we focused on reducing the size of the LM. We pruned the number of *n*-grams in the LM by using a threshold on the relative change in the entropy based on the approach described in [7].

IV. ENGLISH ASR CONFIGURATION

A total of 36 hours of transcribed acoustic data was available to us for estimating acoustic models for English ASR. 13 hours are from the role-playing 2-way data collected under the DARPA TransTac program and 23 hours are from the Wall Street Journal (WSJ) corpus. We first estimated acoustic models using Maximum Likelihood (ML) estimation and later performed lattice-based discriminative training using the Maximum Mutual Information (MMI) [8] criterion. For the handheld system we estimated PTM models and for the laptop system we estimated more detailed STM models.

In addition to the transcripts for the 2-way acoustic data, 90K variants for about 500 questions are available for LM training. For the handheld system, we constructed a dictionary exclusively from in-domain data, resulting in a dictionary of 5K words. We also trained a trigram LM from 1 million words from the in-domain data.

For the laptop system, we first augmented the vocabulary from the in-domain data with frequent words occurring in the conversational telephone speech (CTS) data [9], resulting in a dictionary size of 8K words. We estimated the LM by interpolating the in-domain LM with a LM trained with 49 million words from out-of-domain data consisting of English CTS data.

Table 1 compares the accuracy of the laptop and handheld system on the TransTac March 2006 offline evaluation data prepared by NIST. As shown in Table 1, the number of Gaussians in the PTM model used in the handheld system is a factor of 35 smaller than the number of Gaussians used in the laptop system. The degradation in the WER for un-adapted decoding on the handheld is less than 15% relative over the un-adapted decoding on the laptop system.

We also measured the processing speed of our small footprint English ASR on the March 2006 offline evaluation data. The processing speed on the P2 was 1.1 times real-time (xRT) and 0.6xRT on the HP iPAQ.

Table 1: Comparing English ASR performance on laptop and handheld.

System	Acoustic Model	#Gauss.	%WER
Laptop	MMI STM	108K	12.0
	+ adapt.		5.8
Handheld	MMI PTM	3K	13.5

V. IRAQI ASR CONFIGURATION

A total of 215 hours of transcribed colloquial Iraqi audio data [10] from two different types of data collection is available for acoustic modeling. First, we used the "Grapheme-to-Phoneme" mapping approach first introduced in [11] to generate pronunciations for the Iraqi words. Next, we estimated acoustic models using MMI criterion. For the handheld system we estimated PTM models with 64 Gaussians for each phoneme, whereas for the laptop prototype we used SCTM models with 64 Gaussians per codebook for each state cluster.

In Table 2, we compare the performance of the PTM models with that of SCTM models trained for the laptop prototype. The WER is computed on 9 hours of test data held-out from the corpus. These experiments used a 52K dictionary with 48.5K words from the domain data and 3.5K frequent words from out-of-domain conversational Iraqi data. The LM is trained on 1 million words from in-domain data and 500K words from the out-of-domain data. The degradation for using PTM models in both passes is large compared to using SCTM models. However, the number of Gaussians in the PTM model is a factor of 70 smaller than the SCTM model.

Table 2: Comparing Iraqi acoustic model used in the laptop with the one used on the handheld.

System	Acoustic Model	#Gauss.	%WER
Laptop	MMI SCTM	175K	33.7
Handheld	MMI PTM	2.5K	42.9

Due to the morphological complexity of Iraqi Arabic, the vocabulary size required for ensuring a low out-of-vocabulary rate is significantly higher for Iraqi than for English. It is impossible to fit the 52K word Iraqi ASR system as-is along with the other modules within the limited memory available on the handheld. Therefore, we explored two approaches for developing an Iraqi ASR system for handheld computers.

In the first approach, we explored effective pruning of the vocabulary and the *n*-grams in the Iraqi LM. We found that we could reduce the vocabulary to 10K words with a 3.6% absolute increase in the WER. For the 10K vocabulary system, we further pruned the bigrams and trigrams using entropy-based pruning. Table 3 shows that we can reduce the number of trigrams by almost of a factor of 3 with a 1.7% absolute degradation in the WER.

In the second approach, we exploit the inflective nature of the Iraqi Arabic by developing a hybrid word and morpheme recognition system. First, we identified 65 prefixes and 62 suffixes from the Iraqi training corpus, and used them to decompose words into morphemes (prefixes, stems, and suffixes). Next, we trained an LM consisting of 5K frequent words and 5K frequent morphemes. As shown in Table 4, the mixed word and morpheme system increases the "effective" vocabulary of the Iraqi ASR engine without a significant increase in actual vocabulary size. It also outperforms a similar sized (10K) word system in terms of the WER and is only 1.7% absolute worse in WER, when compared to the full 52K vocabulary system. Note that all acoustic models used in these experiments were PTM models estimated using MMI.

Table 3: Effect of entropy pruning of n-grams on perplexity and WER on held-out Iraqi test data.

Thresh	#tri-grams	#bi-grams	Perp.	%WER
0	516K	279K	96	46.6
1e-10	331K	240K	100	46.8
1e-08	147K	144K	126	48.3

Table 4: Comparing hybrid word and morpheme ASR with word only ASR on held-out Iraqi test data.

ASR System	%OOV	%WER
52K Word	2.2	42.9
10K Word	9.4	46.6
5K Word + 5K Morpheme	1.2	44.6

In Table 5, we compare the WER obtained on the TransTac March 2006 offline evaluation data for the 10K word system, the 10K hybrid word and morpheme system, and the full vocabulary laptop system. The laptop system used STM models in the forward decoding and SCTM models in the backward decoding. The LM on the handheld was pruned using a threshold of 1e-07 on the relative change

Table 5: Comparing laptop and handheld Iraqi ASR configurations on the TransTac 2006 March offline test data.

System	%WER
Laptop: Full vocab. w/o adaptation	28.7
Laptop: Full vocab. with adaptation	23.0
Handheld: 10K word	37.8
Handheld: 10K hybrid word-morpheme	36.4

in entropy. Note that unlike the English system, the WER is significantly worse for Iraqi Arabic on the handheld when compared to the laptop system. We believe that the main reason for a larger degradation on the Iraqi data is due to the high perplexity and relatively higher speaker variability in the Iraqi corpus. The large vocabulary nature of Iraqi and the absence of short vowels are also likely contributors to the higher WER.

In the 2006 March evaluations we used the 10K word Iraqi ASR as the hybrid word-morpheme system was still under development. The processing speed of our 10K word Iraqi ASR measured on the offline evaluation data was 1.9xRT on the P2 and 0.9xRT on the iPAQ. The real-time factor for Iraqi ASR on the laptop was 0.3xRT.

VI. TRANSLATION CONFIGURATION

The end-to-end translation system uses a question canonicalizer [4] for mapping the recognized English speech to a question supported by the system. For translating Iraqi to English, we use our concept translation [4] engine. We integerized both the canonicalizer and the concept translation engine for optimal performance on the StrongARM processors. Since the integerization process did not result in any degradation in performance, we now use the integerized version of the canonicalizer and the concept translation engines in our laptop prototype as well. Thus, there is no performance difference in the concept translation and question canonicalization between the laptop prototype and the handheld. However, the laptop prototype has an SMT fallback in either direction which is not currently available on the handheld. Details on the performance of the concept translation and question canonicalizer are presented in [4].

VII. CONCLUSIONS AND FUTURE WORK

In this paper we have described the ongoing work in developing a bilingual Iraqi/English speech-to-speech translation system for handheld platforms. In addition to presenting algorithms and experimental results for optimizing ASR performance on StrongARM integer processors, we have described a novel engineering solution for integrating the components of the S2S system within the limited amount of memory available on the handheld computers and the constraints imposed by the Windows CE operating system.

Our focus now is to extend the 1.5-way speech translation system on handheld computers by incorporating statistical machine translation and to combine it with concept translation as we have done for our laptop prototype. We will also incorporate speaker adaptation in our handheld system to further improve our ASR performance.

REFERENCES

[1] D. Stallard et al., “Design and Evaluation of a Limited two-way Speech Translator,” *Proc. EUROSPEECH*, ISCA, Geneva, Switzerland, pp. 2221-2223, Sept. 2003.

[2] A. Waibel et al., “Speechalator: Two-way Speech-to-Speech Translation on a Consumer PDA,” *Proc. EUROSPEECH*, ISCA, Geneva, Switzerland, Sept. 2003.

[3] B. Zhou, D. Dechelotte, and Y. Gao “Two-way Speech-to-Speech Translation on Handheld Devices,” *Proc. ICSLP*, ISCA, Korea, Oct. 2004.

[4] D. Stallard, F. Choi, K. Krstovski, P. Natarajan, R. Prasad, and S. Saleem, “A Hybrid Phrase-based/Statistical Speech Translation System,” *Proc. ICSLP*, ISCA, Pittsburg, PA, Sept. 2006.

[5] L. Nguyen and R. Schwartz, “Efficient 2-pass N-best Decoder,” *Proc. EUROSPEECH*, ISCA, Rhodes, Greece, Sept. 1997.

[6] J. Davenport, R. Schwartz, and L. Nguyen, “Towards a Robust Real-Time Decoder,” *Proc. ICASSP*, IEEE, Phoenix, AZ, March 1999.

[7] A. Stolcke, “Entropy-based Pruning of Backoff Language Models,” *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Landsdowne, VA, 1998.

[8] P. C. Woodland and D. Povey, “Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition,” *Computer Speech and Language*, Vol. 16, pp. 25-47, 2002.

[9] R. Prasad et al. “The 2004 BBN/LIMSI English Conversational Telephone Speech Recognition System,” *Proc. EUROSPEECH*, ISCA, Lisbon, Portugal, Sept. 2005.

[10] S. Saleem, R. Prasad, and P. Natarajan, “Colloquial Iraqi ASR for Speech Translation,” *Proc. ICSLP*, ISCA, Pittsburg, PA, Sept. 2006.

[11] J. Billa et al., “Audio Indexing for Arabic Broadcast News,” *Proc. ICASSP*, IEEE, Orlando, FL, May 2002.