Optimizing Components for Handheld Two-way Speech Translation for an English-Iraqi Arabic System

Roger Hsiao[1], *Ashish Venugopal*[1], *Thilo Köhler*[2], *Ying Zhang*[1,2], *Paisarn Charoenpornsawat*[1]
Andreas Zollmann[1], *Stephan Vogel*[1], *Alan W Black*[1], *Tanja Schultz*[1] *and Alex Waibel*[1]

InterACT[1], Carnegie Mellon University, Pittsburgh, PA 15213
Mobile Technologies LLC[2], Pittsburgh, PA
{wrhsiao, awb, tanja}@cs.cmu.edu

Abstract

This paper described our handheld two-way speech translation system for English and Iraqi. The focus is on developing a field usable handheld device for speech-to-speech translation. The computation and memory limitations on the handheld impose critical constraints on the ASR, SMT, and TTS components. In this paper we discuss our approaches to optimize these components for the handheld device and present performance numbers from the evaluations that were an integral part of the project. Since one major aspect of the TransTac program is to build fieldable systems, we spent significant effort on developing an intuitive interface that minimizes the training time for users but also provides useful information such as back translations for translation quality feedback.
Index Terms: English-Iraqi Speech Translation, Handheld Devices, Iraqi Speech Recognition, Translation Interface, Pocket Translation.

1. Introduction

Over the past few years, we have seen substantial improvement in automatic speech-to-speech translation systems not just in the large scale, but also on personal handheld devices. Since our first implementation of speech translation on a consumer PDA in the context of the Babylon program (Speechalator [1]), we were able to substantially improve the quality of computer mediated communication.

As part of the DARPA TransTac program, 6 developers have built portable systems for English-Iraqi speech communication. Some groups developed a two-way system, few groups managed to build the system on consumer PDAs. Here we describe our own work on producing a field usable system on a standard consumer PDA (HP iPaq 2755). Our system runs on most platforms, high-end Pocket PC/Windows Mobile machines and on laptops, under both Windows and Linux. We have also successfully used VoxTec's P2 Phraselator hardware, a ruggedized PDA which offers a built-in audio I/O that is more appropriate for field situations than off-the-shelf PDAs.

In addition to English-Iraqi, which most of this paper is about, the system also supports English-Thai (based on the system described in [2]) and English-Spanish.

Although many will say making speech translation work on standard consumer PDAs is a challenging task, apart from porting the software, the computation and memory limitations of these devices require a substantial redesign of parts of the algorithms and data structures. Our core engines for ASR, SMT and TTS have all had substantial work to make them efficient on a machine with limited memory, slow "disk", and no floating point hardware. Our models, trained on desktop machines, have to appropriately be tuned (in vocabulary, beam width, etc), in order to work well on the target platform.

As evaluations with typical end users have shown, high quality components alone are not sufficient to make a device usable in the field, issues in user interface, microphones, training of users are critical in making a successful two-way portable speech-to-speech translation system.

This paper is organized as follows. In section 2, an overview of our system is provided and we will discuss our user interface. Section 3 is about the ASR. Some considerations for speedup on PDA are also covered. SMT decoder and its training procedure are described in section 4. In section 5, the TTS module in our system is introduced. A summary of our work is provided in section 6.

2. Intuitive Field-System Interface

Since the translation system is designed to be used in tactical environments and should not require too much training, we developed an intuitive user interface. To translate an utterance, the user simply presses a button on the device, speaks the utterance, which gets translated and spoken in the target language. The user can switch between automatic and verification mode. In the former mode, the system automatically translates the spoken utterance and plays the synthesized output to the recipient. In the latter mode, the system asks the speaker for verification of the translation. Only if the speaker confirms the translation, the system presents the synthesized output. To enable users to tell about the translation quality in the other language (that they normally do not speak) we added the functionality of "back translation". Here, the translation of the target language is translated back into the source language. By comparing the original spoken utterance/ASR output with the back translation, the user can judge on the translation quality. Finally, the interface supports logging of all recordings and corresponding component output for later investigation.

Figure 1 shows a screenshot of the Graphical User Interface (GUI) of the system. The GUI window is divided in two boxes, the upper one shows English the lower one shows Iraqi text. These boxes can be populated by either the recognized speech output (ASR), translation output (SMT), or by directly typing in text using the virtual PDA keyboard. This last option is very convenient, e.g. for correcting eventual ASR errors before translation. The "translate" buttons on the touch screen can be used to initiate translation of text in the corresponding box. The "speak" button synthesizes the content of the text box. The button labels "o" is a software record button which can be used if no hardware buttons for recordings are supported by the device. The most convenient usage however, hardware buttons are recommended. In user studies and evaluations, we found that the most intuitive use is to run the PDA in a walkie/talkie modus, i.e. the user pushes and holds one

Figure 1: Screenshot of the English/Iraqi speech-to-speech system.

recording button while speaking. The two languages are assigned different buttons. The software allows for easy button configuration. Decoding of the recording can be stopped at any time by re-pressing the button, which turned out to be beneficial since in case of unintended input, the user does not have to wait until the speech is processed before starting over. The GUI indicates the system's state by color-coding squares, in waiting state, the square shows the signal level of mic input, while recording the square turns red, decoding is shown by yellow and green squares for ASR and SMT.

3. Automatic Speech Recognition (ASR)

As the main input modality, we implemented speaker independent speech recognition using the integrated microphone of the mobile device. The ASR system uses the Janus Recognition Toolkit (JRTk) featuring the IBIS decoder [3] for the laptop version, and a PDA-optimized port [4] for the PDA version, that runs around 2-5x real-time on the 624MHz Intel XScale PXA270 processor. This is the standard processor found in most commercial PDAs. Because the ASR runs over the utterance in one pass, the recognition can immediately begin after the user starts recording. Adaptation steps, that are needed for the Feature Space Adaptation (FSA) or Maximum Likelihood Linear Regression (MLLR), are processed hidden for the user during the synthesized voice is speaking. However, we use adaptation in our laptop version only, because on the PDA the required calculations are either too slow or too inaccurate for reliable performance improvements.

3.1. English ASR

The English ASR system is a three-state sub-phonetically tied semi-continuous recognizer composed of 2000 context dependent distributions with as many codebooks. Each codebook has 16 Gaussians and it takes 32 dimensional Mel Frequency Cepstral Coefficients (MFCC) after linear discriminant analysis (LDA) as input. The acoustic model was trained on the Hub4 BN data and meeting transcriptions from ICSI, NIST and CMU [5]. It uses a trigram language model with approximately 600K trigrams and a vocabulary size of 8K words. The language model was trained on several text corpora in force protection domain, which sum up to 1.7M words. A subsequent interpolation with a medical and touristic language model of a lower weight was performed, in order to gain a wider coverage of vocabulary and domain.

The laptop version achieves a WER of 8.8% on the TransTac March 2006 evaluation data. The PDA version uses the same language and acoustic models, but has tighter search beam settings and no incremental FSA or MLLR adaptation. It uses several speedup techniques (described in 3.3) that result in an overall WER of 14.6%. The TransTac March 2006 evaluation data consists of 2880 utterances for each English and Iraqi. Each language has around 3 hours of audio data.

3.2. Iraqi ASR

The acoustic model uses the same topology of English ASR. It consists of 2000 codebooks with 32 Gaussians, hence, it is larger than the English ASR model. The feature extraction process remains the same. The acoustic model is trained with around 93 hours of Iraqi speech data including data sets from Appen/BBN, Cepstral, IBM/DLI Pendleton, and Marine Acoustics Inc. A laptop version is built using the same data but larger codebook size of 3000 with 32 Gaussians.

The language model for Iraqi ASR is a smoothed trigram model using modified Kneser-Ney method [6]. The training set consists of 1.2M words, including data from different domains in force protection and medical process such as, common community interest, medical screening, traffic control point, and some less restricted topics, such as rapport building. The vocabulary is based on frequency count and the size is around 7K words.

Compared to the English ASR, there is less data for the Iraqi language model. To improve performance and have broader coverage, we explored the possibility of incorporating modern standard Arabic (MSA) data into the language model. Arabic Gigaword second edition is an archive of Arabic newswire data and it was used to train a trigram language model. The resulting model is then interpolated with the Iraqi language model.

Based on the Iraqi vocabulary, 115M words were sampled from Gigaword corpus by considering whether the sentence has more than 75% of words which are in the 7K words Iraqi vocabulary. It is then interpolated with the Iraqi language model by using some heldout data. Table 1 is a summary of perplexity reduction by using the interpolated language model.

Table 1: Relative perplexity reduction for different scenario.

	# tokens	Iraqi only	Iraqi+MSA	Rel. imprv.
CCI	56K	295.01	281.79	4.48%
SWET	46K	486.63	460.94	5.28%
Search	49K	226.28	224.45	0.81%
TCP	49K	305.23	301.76	1.14%
Medical	21K	368.83	350.63	4.93%
Joint	11K	412.49	355.24	13.88%
RB	8K	354.77	337.06	4.99%

- CCI: common community interest.
- SWET: about sewage, water, electricity and trash.
- Search: searching for people, houses and weapons.
- TCP: conversations with drivers at a traffic control point.
- Medical: about medical attention.
- Joint: joint mission with Iraqi police or special forces.
- RB: rapport building.

This experiment is based on the data collected by DLI and the data is divided into different categories as described in table 1. The interpolated model has mild improvement on different scenarios, and the improvement differs a lot on different topics. It may suggest MSA data can be helpful for certain domains in Iraqi speech recognition. However, in overall, we found that the system achieves the same WER by using the interpolated model. We decided to use the interpolated model in order to improve the coverage, but to reduce the size of the language model, only 550K words of MSA data were used in the PDA system.

The PDA system achieves a WER of 34.0% on the TransTac March2006 evaluation, and the laptop system has a WER of 15.4%. On a test set based on Cepstral, Appen/BBN and Marine Acoustics data in force protection domain, our PDA system scores 27.7% WER. This test set consists of around two hours of audio data and 2246 Iraqi utterances.

3.3. PDA Specific Speedup Techniques

The processor clock of our target mobile devices at the time this paper is written ranges from 400 to 624MHz, with a small memory cache of 64KB and without floating point hardware. Compared to a laptop or a desktop machine, a simple recompile of the ASR system for the PDA results in an almost unusable system, mainly because of the emulated floating point operations, but also because of the slow CPU and memory access.
To compensate the limited resources of the PDA, several speedup and memory saving techniques are implemented, while keeping the same acoustic and language models.

All CPU intensive preprocessing steps are performed using integer arithmetic, in order to avoid software emulation of floating point calculations, which typically take 10 to 20 times longer than their integer equivalents. Functions that benefit from this kind of optimization are the FFT and the matrix multiplication, and also the calculation of Mahalanobis distances in the acoustic model during decoding. In general, the integer calculations do not seem to harm the performance significantly, if a proper pre-scaling and overflow protection is applied.

A Gaussian selection algorithm is used to speed up the evaluation of the acoustic model. As a small modification compared to [7], we generated the disjunct clusters using Euclidean distance, and used Mahalanobis distance between a feature vector and the cluster centroids to find the active clusters for each frame during decoding. The covariance of each centroid is calculated by averaging over all Gaussians' covariances that belong to the centroid's cluster. In our system, 128 clusters were used and only the Gaussians in the top 25% of best matching clusters were evaluated for each frame.

The Early Feature Vector Reduction [4] (EFVR) is used to remove redundant consecutive feature vectors, as found in silence and static vowels or noises, which results in a reduction of 25 to 50% of the feature vectors before they are fed into the decoder.

The Gaussians (means and covariances values) of the codebooks are compressed from 32-bit floating point values to 8-bit integers, saving 75% of the memory to store the acoustic model.

4. Statistical Machine Translation (SMT)

We used a statistical machine translation (SMT) system as the translation component in the S2S system. The decoding algorithm used in the SMT decoder follows the CMU SMT system [8]. The translation model leverages underlying technology developed in [9] with morphological splitting techniques from [10] implemented to improve English to Iraqi translation. Parameters of the SMT system were trained on development data from the 2005 DARPA TransTac evaluation as well as selections from the spontaneous speech component of the training data.

4.1. Statistical Machine Translation Decoder

The SMT decoder was re-engineered for PDA by rewriting the program for the Windows CE operating system. More importantly, we have designed compact data structures for the language model and the translation model specifically for the PDA to make a two-way phrase-based translation system fit into the limited memory and translate in real-time. To make the decoder running efficiently on the PDA, only monotone decoding is used.

Similar to the situation in ASR, we have to use integers for all the probabilities since there is no floating point coprocessor in the PDAs. A pilot study on the standard decoder showed that the overall translation quality did not degrade when converting all the probabilities from float/double to integers.

All the words used in the translation/language models are mapped into unique integer vocIds. Two-byte integers are used for vocId and the system can handle a vocabulary of 64K words. The transcription output from the ASR will first be mapped from text to a sequence of vocIds, and vocIds are used during the decoding to avoid the hassle of mappings. After the decoding, VocIds of the translation result are mapped into words for the TTS module.

The n-gram language model is converted from its text representation to the binary compacted format. By doing this, each n-gram is represented by n vocIds and a fixed number of bytes for conditional probabilities and back-off weights. We store all the n-grams in a binary file in sorted order. The decoder can directly look up an n-gram for its information in the file without loading the language model into the RAM. This saves the limited RAM for other programs such as the decoding process which requires random access of dynamic data structures.

The translation model requires most of the memory in a SMT system because the number of phrase pairs can easily go up to millions. As we are doing two-way translation, there are two translation models involved: Iraqi to English and English to Iraqi. Both models have English and Iraqi phrases. To avoid the redundancy of saving one phrase twice, all English phrases used in both translation models are sorted and stored in one binary file, and the same for all the Iraqi phrases. Each English(Iraqi) phrase is then mapped to an integer "phrase ID" which is used to convert the translation models into a binary file. Similar to the language model, the decoder can access the translation model directly from the binary file instead of loading it into the RAM. Thus in the SMT module, only the decoder search process requires memory from the RAM.

In our implementation, the translation models can have up to 1M unique English phrases and 1M Iraqi phrases and 16M phrase pairs for each translation direction.

With these PDA-specific design and other tailored implementation in the decoder, the translation speed is impressively fast. On average translating a sentence takes less than 10 ms.

4.2. Training for SMT

While our desktop based systems support multiple floating point translation model scores for each phrase translation, our PDA based decoder operates with a single integer range score. Our traditional SMT system [9] uses six translation models scores and an additional phrase counter. We collapse these scores into a single score by taking the dot product of each score with their respective scaling factors trained by MER [11]. We then adjust all the scaling factor to limit the number of entries that are above the converted integer range.

The parallel data for the SMT component used approximately 20K parallel sentences representing 670K words of Iraqi and 982K words of English. The majority of this data (18K sentence pairs) are transcripts from spontaneous speech interactions, while the remaining data are primarily transcribed from English to Iraqi question answer sessions.

Iraqi to English SMT models are trained on parallel corpora in the Force Protection and Water Electricity and Sewage domains. Unlike typical parallel corpora used for SMT, the Iraqi data consists mainly of question/answer and command/response exchanges, many of them collected *in situ*. As a result, the content and nature of English sentences that have been translated into Iraqi,

are distinctly different from sentences that were originally spoken (and then transcribed) into English. We account for this difference within the language model, by interpolating a model built on all the *target* (English for Iraqi→English system) with the subset of the data that has been translated from source to target language.

4.2.1. Iraqi to English

Iraqi, like most Arabic dialects exhibits significant morphological inflection, which is a cause for concern when building statistical translation models.

We address issues in Iraqi morphology by applying the techniques presented in [10] to split Iraqi surface forms into multiple segments that are better suited to alignment to English surface word forms. Once the alignment models are built, we use the techniques described in [9] to extract phrase translations from the parallel corpus. Since we built our models on the fragmented Arabic morphological elements, we map them back to their surface form (by storing a lookup table at the time of initial splitting). These corrections yield significant improvements in score, approximately 4.8% on the NIST metric resulting in a test set score on the 240 sentence December 2005 TransTac Eval of 7.05 (when MER optimized with development data sampled from the parallel corpus).

4.2.2. English to Iraqi

The English to Iraqi direction presents challenges in translating into a morphologically rich language. We primarily mitigate this issue by allowing a larger portion of the final phrase table to represent phrases from English to Iraqi and changing the parameters for the phrase extraction. In phrase extraction, we set the extraction parameters in [9] to those calculated after 3 iterations of MER training that began with default parameters (2.0, 1.0, 0.5 in both directions) selecting relatively shorter Iraqi phrases as determined by the MER optimal parameters. This step increases the NIST score on the December 2005 Evaluation data from 5.0 to 5.68.

5. Text-to-Speech (TTS)

The text to speech component uses the Swift engine from Cepstral LLC, with English and Iraqi Arabic voices. The English voice is around 20M and the Iraqi voice around 12M. The Iraqi dialect has the addition issue that it is not normally written. Thus a standard for writing, using standard Arabic script, Iraqi was defined for the whole TransTac project. However as that standard did not include explicitly vowels in the text (as is common in MSA), and that we only have a complete pronunciation lexicon for one large subset of the data, we had to augment our pronunciation models with a statistical method for predicting the pronunciation of unknown words. We built statistical models using our letter-to-sound CART tree methods [12]. On held out data of 12K words, this technique achieves 52.31% words correct and 86.51% letter correct. We use this same pronunciation model to construct the lexicon for the ASR.

6. Conclusions

In this paper, we have presented our two-way handheld speech-to-speech translation system in limited domain. We found that handheld devices have severe constraints on computation and memory, and these issues have to be considered in optimizing the ASR, SMT and TTS engines.

User interface is another important factor for a successful device. It needs to be simple and clear so that the time for training a new user is minimal. Our system adopts a push-to-talk mechanism, a simple layout and a back-translation scheme which allow people to use our system reliably in a few minutes.

7. Acknowledgements

This work is in part supported by the US DARPA under the program "Spoken Language Communication and Translation Systems for Tactical Use (TRANSTAC)", under a grant entitled "An Iraqi-English Two-way PDA-based speech translator." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

8. References

[1] A. Waibel, A. Badran, A. Black, R. Frederking, D. Gates, A. Lavie, L. Levin, K. Lenzo, L. Mayfield Tomokiyo, J. Reichert, T. Schultz, D. Wallace, M. Woszczyna, and J. Zhang, "Speechalator: two-way speech-to-speech translation on a consumer PDA," in *Proceedings of the European Conference on Speech Communication and Technology*, 2003.

[2] T. Schultz, A. Black, S. Vogel, and M. Woszczyna, "Flexible speech-to-speech translation systems," *IEEE Transactions in Speech and Audio Processing*, vol. 14, no. 2, pp. 403–411, Mar. 2006.

[3] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one pass-decoder based on polymorphic linguistic context assignment," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001.

[4] T. W. Köhler, C. Fügen, S. Stüker, and A. Waibel, "Rapid porting of ASR-systems to mobile devices," in *Proceedings of Eurospeech, 9th European Conference on Speech Communication and Technology*, 2005.

[5] F. Metze, C. Fügen, Y. Pan, T. Schultz, and H. Yu, "The ISL RT-04S meeting transcription system," in *Proceedings of the NIST RT-04S Evaluation Workshop, Montreal, Canada, May 2004, NIST*, 2004.

[6] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, Arivind Joshi and Martha Palmer, Eds., San Francisco, 1996, pp. 310–318, Morgan Kaufmann Publishers.

[7] J. Leppnen and I. Kiss, "Gaussian selection with non-overlapping clusters for ASR in embedded devices," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[8] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venogupal, B. Zhao, and A. Waibel, "The CMU statistical translation system," in *Proceedings of MT Summit IX*, New Orleans, LA, September 2003.

[9] S. Vogel, "PESA: Phrase pair extraction as sentence splitting," in *Proceedings: the tenth Machine Translation Summit*, Phuket, Thailand, 2005.

[10] A. Zollmann, A. Venugopal, and S. Vogel, "Bridging the inflection morphology gap for arabic statistical machine translation," in *Short Papers in the Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, 2006.

[11] F. Josef Och, "Minimum error rate training in statistical machine translation," in *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7 2003.

[12] A. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *Proc. ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia., 1998, pp. 77–80.