Speechalator: two-way speech-to-speech translation on a consumer PDA

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]
Jürgen 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 paper describes a working two-way speech-to-speech translation system that runs in near real-time on a consumer handheld computer. It can translate from English to Arabic and Arabic to English in the domain of medical interviews.

We describe the general architecture and frameworks within which we developed each of the components: HMM-based recognition, interlingua translation (both rule and statistically based), and unit selection synthesis.

1. Background

As an initial part of the DARPA Babylon project we were tasked with building an interlingua-based two-way speech-to-speech translation system on a small device in a language that our group had no (significant) previous experience in. This required us to solve three specific problems:

- How to collect sufficient and appropriate data for translation, recognition, and synthesis, in the most efficient way. Using foreign language experts we designed protocols to define the translation domain and collect examples to allow an appropriate interlingua to be designed.
- How to take advantage of both knowledge-based techniques in defining an interlingua; and statistical techniques, in learning the relationship between surface forms and that interlingua; in such a way to make transfer to new domains and languages efficient.
- How to fit two recognizers, two synthesizers and a two-way translation system on a device with only 40Mb of available space and limited CPU power. This required addressing engineering issues: lack of floating point support, synthesis database compression, efficient recognition decoding algorithm; as well as research issues in model design for size and efficient access.

The end result is a working prototype on a Compaq iPaq which can recognize, translate and synthesize bi-directionally between two languages, English and Egyptian Arabic, and do so in a reasonable time. Although this prototype is limited, it was aimed at medical interviews, and deals with only many hundreds of sentence types, it shows the feasibility of such a system.

This particular system was built over a period of six months, using the tools and techniques we have developed over a number years in rapid development for speech-to-speech translations systems [1], [2].

2. Data Collection

The Arabic language was mostly new to this group. Although we had some experience and resources from Tunisian Arabic in recognition as part of the GlobalPhone project [3], we were essentially starting in a new language. This was a good test of our existing speech-to-speech translation framework.

The normal Arabic script does not include all vowels, although there are diacritics which can be used to specify all vocalization, these only appear in childrens' books and the Koran and do not appear in conventional text. Thus normal script would be hard to use for speech processing. There are statistical techniques (e.g. [4]) that can be used to predict vocalization but it is much easier if we could use a script with all vowels fully specified. There have been successful attempts to do Arabic speech recognition without explicit vowels [5], but for synthesis this would be much harder if actually possible. Therefore because we are embedding this use of Arabic within speech-to-speech translation where we control in the input and output mechanisms we are in a position to stipulate that the internal form may be a romanization which contains full vocalization. Transliterating from a romanized script into Arabic script is easy (it involves removing information) so we can still display the translation in Arabic script, but internally preserve the vowel information. Others have noted this problem and we based our romanization on the Arabic CallHome [6] romanization, but made several refinements, from which phonetic forms can be easily derived.

The second major issue was what dialect of Arabic to use. Although there is a standard written form for Arabic, Modern Standard Arabic, (MSA) this is not used for normal conversation. As we are specifically interested in *spoken* language translation we decided to chose a major spoken dialect for which local experts were available. Thus we settled on Egyptian Arabic.

There were three areas for which data had to be collected: recognition, synthesis and translation. From an existing database of English medical expressions used for another speech-to-speech translation system, Arabic foreign language experts (FLE) hand translated each utterance into a number of different paraphrases in Egyptian Arabic (up to 10 different examples). The FLEs were then asked to speak each of the utterances. So that we collected recordings of some 7500 in-domain utterances with romanized transcriptions.

3. Recognition

Speech Recognition is the most computationally expensive part of the speech-to-speech translation process. Unless a decoder is specially designed to run on a PDA platform which has limited memory bandwidth and no floating point, the recognition will likely be too slow for practical use.

Multimodal Technologies Inc, has been working on a small footprint fast decoder HMM-based recognition for some years and has had significant experience in working with multiple languages and speech-to-speech translation systems.

The audio input device on PDAs is not of high quality. Given the size of the hardware it is common that the audio channel has lots of electrical noise from the power supply and motherboard, thus recordings on these devices are not clean. Furthermore in our experience the amount of noise that the audio channel may differ from device to device. External digitizing of audio might be an option in the long term, such as off-device USB audio, or design of better shielding around the audio hardware, but our goal was to use standard PDAs so such alternatives were not available.

The acoustic models were bootstrapped from the GlobalPhone [3] Arabic collection as well as the recordings described above. The data contains both male and female examples though we have tested more with male speakers than female.

As the Speechalator is a domain-based translation system, we want to use that advantage to constrain the recognition engines. Rather than having a separate language model and subsequent parser as we have done in other translation systems we have built [7], we have integrated the parsing part of the system within the recognizer language model. This allows the decoder to be more efficient allowing us to deal with larger vocabularies and more utterance types than we would be able to do otherwise.

The final part of the recognition system is the adaptation to the acoustic environment, and speaker. This is fairly standard in most recognition engines and we adopt these techniques here too.

4. Translation

Our translation uses an explicit language-independent interlingua formalism, so that support of new languages can be achieved without affecting existing supported languages. Design of the interlingua formalism is not easy but we already have experience in that area, [8].

Our interlingua representation is based on speaker intention rather than literal meaning. The speaker's intention is represented as a domain-independent speech act followed by domain dependent concepts. We use the term *domain action* to refer to the combination of a speech act with domain specific concepts. Examples of domain actions and speech acts are shown in Figure 1. Domain actions are constructed compositionally from an inventory of speech acts and an inventory of concepts. Specific information concerning predicate participants and objects etc. is represented by arguments and values. The allowable combinations of speech acts, concepts, arguments and values are formalized in a human- and machine-readable specification document.

Our initial system used an off-device interlingua to text generation system as the generator had not yet been ported to the PDA device. This worked well, but given the network overhead, was slower than we wished, but could deal with large grammars.

We took two parallel tracks to solve this, this first was to investigate porting the existing generator system to the PDA, but as it was in C++ that was going to take time. The second route was to use a statistical based translation mechanism.

I have a husband and two children ages two and eleven.

```
give-information+personal-data
 (family=
   spec=(conj=and,
         (spouse, sex=male),
         (offspring,
          quantity=2,
          age=(quantity=(conj=and,2,11)),
  experiencer=i)
```

Do you have any pain in your arm?

```
request-information+experience+health-status
 (health-status=pain,
  body-location=arm,
  experiencer=you)
```

I could examine your shoulder.

```
offer+give-medical-procedure+body-object
 (who=i,
  medical-procedure-spec=medical-examination,
  body-object-spec=(whose=you, shoulder))
```

Figure 1: **Examples of Speech Acts and Domain Actions**

Statistical machine translation has become more popular as its performance has improved. Normally models are trained on corpora of parallel text, with each utterance in one language corresponding to a translation in the target language. This basic model however would remove the advantages of interlingua, as conventional statistical MT techniques would require parallel corpora for each language pair we wished to support. Thus instead of having the two sides be textual utterances we used a parallel corpora of interlingua representations and their realization as textual utterances in the target language. This model does introduce different problems, as the interlingua representation is effectively a tree structure. These techniques are relatively new and will be published elsewhere. But because they were successful and because we had an efficient implementation of the engine on the PDA, it was possible to use this engine with the Speechalator and a fully untethered translation device.

Thus we have two methods, one solely on the device, and a second clear method to cover much larger translation problem if a wireless connect to a server is available.

Only basic evaluation was carried out on these generation models, and this is still continuing work.

5. Synthesis

The speech synthesis was constructed by Cepstral, LLC using techniques that allow high quality unit selection synthesis on small footprint as demanded by the intended platform.

The English male voice (a female voice is also available) offers clear speech in a command-like style. This voice had been created for previous projects and was specifically designed for delivery short dialog utterances such as would be needed in this application.

The voice is a general speech synthesizer that can say anything, and is not limited to a particular domain. At 11KHz,

a suitable sample rate for the PDA hardware, the voice plus the language front end (including the lexicon) is about 9 megabytes.

The Arabic synthesizer was built specially for this project. An initial test voice was built in the Festival Speech Synthesis System [9]. This allowed a certain amount of tuning before a small footprint delivery was attempted.

We used the romanization decided on for the recognizer and translation engine, as predicting vowels in Arabic script is a non-trivial problem. Using the generated list of translations created for the translation part of the system we use a method as described in [10] to select an optimal subset of these utterances that best cover the acoustic phonetic space. Thus from a list of around 7500 sentences we selected 666 sentences, 102 sentences hand constructed to cover numbers, and 52 two general greetings (for both male and female speakers).

One notable aspect of the building of an Arabic voice for this system was that we found our native speaker slightly reluctant to have their voice used in a device that could later be used by the military for unspecified use, potentially in their own country. With the improvement in speech synthesis to the level where the output voice is recognizable as the particular person who recorded the database, we must be sensitive to the uses of the system we build. Although we are very careful to explain to all our voice talent what the consequences of recording a synthesis voice are, people may not be fully aware until they see the complete system. Because of this, we used a different speaker for the final recordings.

Evaluation of speech synthesis is always hard but there are simple diagnostic tests that can be run to identify problems in the synthesizer. In this case we carried out three specific tests. Based on the Diagnostic Rhyme Test [11], and Modified Rhyme Test we constructed simple mono-syllabic words which differed in one phonetic feature. For English this test typically includes aspects like voicing, nasality, sustenation etc. We modified this list for Arabic and included emphaticness to the class. A second level test involved sentences that were not part of the recorded database but still considered "in-domain". The DRT/MRT and in-domain sentences were then played to native Arabic speakers and they were asked to mark any words which "sounded bad" for any reason, a deliberately vague term. The results are as follows show percentage of "good" words in the synthesized utterances

DRT	MRT	Sentence
78.3	72.0	84.7

These numbers are comparable to English voices, of similar size and degree of development.

6. System Integration

The aim of this work was to deliver two-way speech-to-speech translation on a handheld. The original Babylon project intention was to use an updated platform that was currently used in the one-way Phraselator system [12]. But as that platform was not yet ready we decided to aim for a consumer off-the-shelf (COTS) PDA.

From our experience in other project in delivering on limited hardware platforms we have felt it better to aim for the intended hardware platform at the start rather than assume the platform will improve of the length of the project.

Our first intention was to design a architecture that would allow component to reside either on the device itself or on external servers accessed through a wireless link. Although the ultimate result should not rely on wireless links to server, this architecture would allow us develop and test the components before they were ported to the PDA itself.

Cepstral and Multimodal Technologies had already spent significant effort in produce speech synthesizers and speech recognizers respectively, that were optimized for the StrongARM platform. The process available on most PDAs today is the StrongARM SA1110 (206MHz) or the XScale, used directly as a StrongARM replacement. Neither of these processors are particularly fast and neither offer floating point. Although floating point instructions can be emulated this is far too slow for core functions. Although the Arabic speech models were built just for this projects, and the English speech models were adapted, it was necessary to have already developed the core engines and related model building systems before hand in order to delivery a full two way system in a new language in such a short time.

The engines used for interlingua translation had not yet been ported to Windows CE. Thus at first we used a wireless connection between the PDA and a Linux server to provide the interlingua-to-text part of the both the English and Arabic generation. We later replaced this with an on-device statistical generator that was computationally light enough to run on the device itself. That this statistical generation could be run on the device was not because statistical generation is inherently more less computationally intensive than the rule based generator but that the statistical generator was developed with a port of Windows CE in mind.

The parser part of the system was moved into the recognizer so that the parsing restrictions could better constrain recognition. On such a limited device efficient recognition is important and linking the ASR decoder with a strong appropriate language model is a good thing to do.

The whole system was built as a single binary, being the best use of the process model under Windows CE. Each module maps in its appropriate language data files. Although at present we only have examples in English and Arabic there is nothing language specific in the basic engines.

At run time the system uses around 28Mb of memory and hence can comfortably run on a 64Mb PDA. However as the run-time memory and the storage memory are distinct on most PDAs we also need a separate storage card installed to hold the system (about 30Mb). We have been using Compaq (HP) iPaq 3800 series machines (StrongARM 206MHz) and 3900 series machines (XScale 400MHz) for basic development but also have ported the system to the Dell Axim and the one-way Phraselator hardware. The Dell Axim (XScale 300MHz) has only 32Mb, but we found the system ran well, though slower than on 64Mb machines.

7. Performance

We have not yet, at this stage in the project, been able to run formal evaluation tests, though throughout the six months that the project was active we did carry out component-based tests.

The whole system (running in unthethered mode) takes around 2-3 seconds to translate a typical utterance from when the speaker stops speaking, to when the system starts speaking the translation. Thus, the performance can be said to be just over real-time. However, recognition can take 1-2 seconds longer in adverse acoustic environments.

In spite of PDAs having poor audio input hardware the system works well in various environments, including offices and outside. Though in some harsher environments the system capability improves if given a few utterances to adapt to. We have found that in environments with lots of human speech around,

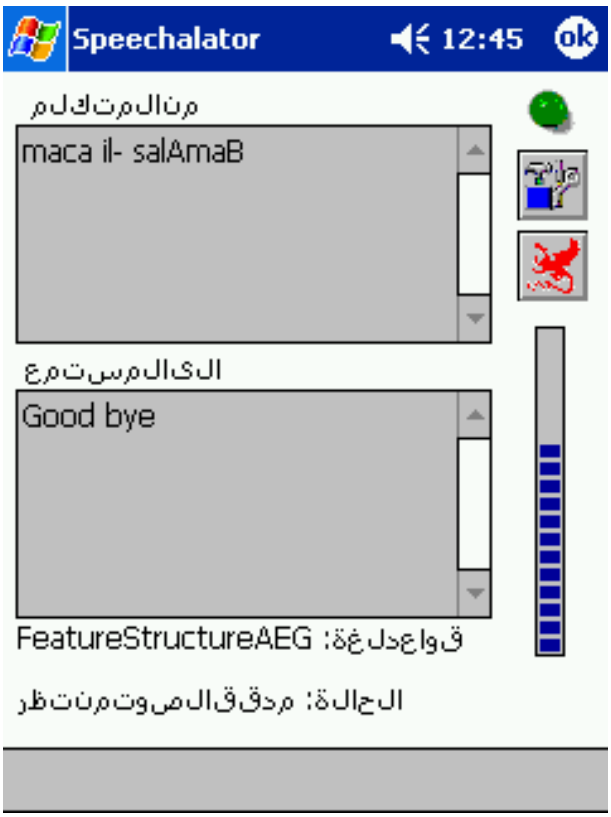
such as bars and restaurants, performance goes down.

In informal tests we have found a greater than 80% accuracy.

The system is set up for the domain of medical interviews, and has only basic vocabulary for greetings and numbers outside that domain. Although there is shared coverage it is assume that the English speaker is a doctor and the Arabic speaker is the patient.

Although difficult to fully quantify the coverage, for English the language model covers many hundreds of sentence types, with as many as dozens of possible variations, such as diseases, ailments and body parts. The Arabic side is more constrained, but still deals with a few hundred sentence types.

8. User Interface



Arabic input Screen
Speechalator snapshot

The user interface is simple, but still requires the users to know something about the operation of the machine. There is a push-to-talk button, and the recognized utterance is displayed the upper window. The second window then displays the translation as it is spoken. The utterance may be repeated at the press of a button. The input language may be change pressing another button. The display uses the native character script for the language that is to be recognized.

There are however usability issues with such a system which we have not yet addressed. We currently have no explicit way of instructing a non-English speaking person in how to use the device. In close conversation the audio output volume is adequate but in larger groups or outside more volume is required than the standard PDA speaker can deliver.

9. Summary

This paper describes a two-way speech-to-speech translation system that runs on a conventional PDA. It translates from English to Arabic and Arabic to English in the medical domain. The system was build over a period of about 6 months. Although we built upon existing engines and techniques, the Arabic language aspects of this work were all carried out within the course of the 6 months.

10. Acknowledgments

This work was partially funded by a grant N66001-00-C-8007 under the DARPA Babylon program: "Mobile Speech-to-Speech Translation for Military Field Application." The opinions expressed in this paper do not necessarily reflect those of DARPA.

11. References

[1] A. Lavie, L. Levin, T. Schultz, and A. Waibel, "Domain portability in speech-to-speech translation," in *HLT2001*, San Diego, California, 2001.

[2] A. Black, R. Brown, R. Frederking, R. Singh, J. Moody, and E. Steinbrecher, "Tongues: Rapid development of a speech-to-speech translation system," in *HLT2002*, San Diego, California, 2002, pp. 2051–2054.

[3] T. Schultz and A. Waibel, "Language independent and language adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, 2001.

[4] K. Kirchhoff, et al., "Novel speech recognition models for Arabic.," Technical report, Johns Hopkins University, 2003.

[5] J. Billa, M. Noamany, A. Srivasta, D. Liu, R. Stone, J. Xu, J. Makhoul, and F. Kubala, "Indexing of arabic broadcast news," in *ICASSP*, Orlando, Florida, 2003.

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

[7] 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.

[8] L. Levin, D. Gates, D. Wallace, K. Peterson, A. Lavie, F. Pianesi, E. Pianta, R. Cottoni, and N. Mana, "Balancing expressiveness and simplicity in an interlingua for task based dialogue," in *Proceedings of ACL 2002 workshop on Speech-to-speech Translation: Algorithms and Systems*, Philadelphia, PA., 2002.

[9] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," http://festvox.org/festival, 1998.

[10] A. Black and K. Lenzo, "Optimal data selection for unit selection synthesis," in *4th ESCA Workshop on Speech Synthesis*, Scotland., 2001.

[11] J. Logan, B. Greene, and D. Pisoni, "Segmental intelligibility of synthetic speech produced by rule," *Journal of the Acoustical Society of America*, vol. 86(2), pp. 566–581, 1989.

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