Portable speech-to-speech translation on an Android smartphone: The MFLTS system

Ralf Meermeier ralf.meermeier@raytheon.com
Sean Colbath sean.colbath@raytheon.com
Martha Lillie martha.lillie@raytheon.com
Raytheon BBN Technologies, 02138 Cambridge, Massachusetts, USA

Abstract

For US troops on the ground in countries like Iraq and Afghanistan, one of the key objectives, ”Winning the Heart and Minds” of the local population, presents a formidable challenge due to the language barrier involved. Employing human interpreters to address the issue has many of its own challenges, foremost availability of locals to willingly act as such. Because of this bottleneck, many of the Army’s humanitarian missions are hindered as they require significant interaction between soldiers and the local population.

The Machine Foreign Language Translation System (MFLTS), a US Army project that originated out of DARPA’s ”TransTac” research effort, aims to address this bottleneck by equipping each soldier with a personal translation device running on a COTS Android smartphone. With it, soldiers can maintain basic free-form conversations with individuals in a turn-based ”radio interview” style, with specific focus on topics such as checkpoints, information gathering and medical help. It can also be operated with optional peripherals that ease the interaction and improve the overall accuracy of the system.

1 Introduction

The paper is structured as follows: In Section 1 we outline the history of the MFLTS program, and in Section 2 we present the distinct challenges that have to be overcome when designing a speech-to-speech (S2S) Android application. Section 3 presents a conclusion that looks forward to where the application could go.

1.1 History of MFLTS

The MFLTS project’s origins can be traced back to DARPA’s ”Translation System for Tactical Use” (TransTac) program, which aimed to spur research in the feasibility of running a full speech-to-speech system on a portable device. Partially in response to earlier systems that worked on the basis of choosing from a fixed set of phrases (and the limitations arising from that), the goal of the research project was to allow for free-form responses from both the soldier and the foreign speaker. Initial prototypes ran on full-fledged laptops which soldiers would carry in a backpack, but once cheap and powerful smartphones entered the market, specifically Google’s ”Nexus One” Android phone (a single-core 1GB ARM device with 512MB RAM), a push was made to transfer the system, and in turn its core technologies (speech recognition, mahchine translatrion, text-to-speech), to this platform.

In terms of language skill, the systems were desired to be at "ILR 1" level (*http://www.govtilr.org*), which corresponds to a person having a basic command of a foreign language, able to understand and pose pertinent questions. With a scale as notoriously difficult to evaluate as this, it is nonetheless the view of the authors that the system exceeds this basic level and is, when used to its full extent, better rated at ILR level 2.

The project had competing teams (BBN, IBM, CMU) build systems that were evaluated in regular intervals at NIST or MITRE (David Stallard, 2011).

Figure 1: Early S2S prototypes on the Nexus One

With the success of the TransTac project, the US Army subsequently created the MFLTS program with the intent of transitioning the research system into a fieldable system that would eventually get deployed in theater. BBN, a consistent top performer in the TransTac evaluations, was chosen to build this framework and reimplement its version of speech-to-speech in it.

The intent of the MFLTS program, however, has much broader scope: To avoid creating a one-off application that would tie the Army to one specific vendor, designing MFLTS as a framework allows easy writing of any application that wants to utilize natural language processing (NLP) components. An application makes a request to the framework for specific NLP components (ASR, MT, etc) and the framework instantiates these components for the app to interact with. The existing framework is massively parallel, adapting to the changing usage patterns, and even is "self-healing", i.e., it will replace crashed components as transparently as possible to the application in order to provide minimal downtime in possibly critical scenarios.

With this framework, the original S2S application is now just a specific app written for the framework. Not only that, but because MFLTS is required to support both Windows and Android operating systems, the same code can be used (with minor adjustments) to run the same application on those vastly different platforms. Another benefit is that the NLP components are designed to be plug-and-play, meaning any third-party vendor can provide new components; as an example, BBN recently replaced its old Byblos recognizer with the new "Sage" recognizer (Roger Hsiao, 2016) (Meermeier and Colbath, 2017) the application, however, has no knowledge of this and simply receives better ASR results. Just as easily, an application's voice output can be changed when a cheaper, or better, TTS provider wraps its engine using the MFLTS API.

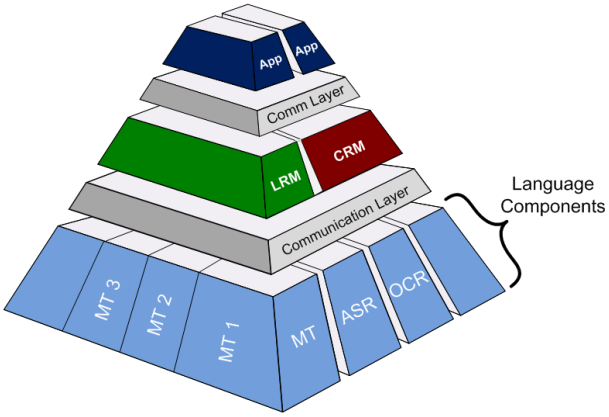

Figure 2: The MFLTS architecture: Apps are connected to NLP components by the framework

1.2 MFLTS S2S

(a) S2S mounted on vest, including peripherals

(b) Application main screen

Figure 3: The MFLTS S2S Android application

In general, the application works in an ”interview-style” fashion where the soldier is the driver of the conversation. Initially he/she will start the conversation by speaking into the application, and then waiting for it to recognize, translate and output the translation via text-to-speech (TTS). All this happens close to real-time, with latencies from end-of-speech to begin-of-TTS on the order of 1000 milliseconds on the Android system. On the much more powerful Windows desktop system there is virtually no latency.

Depending on the conversation, the foreign language speaker (”FLE”) might then be the one responding in turn, just as before speaking into the application and waiting for the translation. It is important to note that it is the soldier who signals through *body language* to the FLE that it is his turn to speak; not using the application to queue the FLE (e.g., with a TTS prompt) was a key realization that later became a design principle of the application.

In its current form, there are two ways to interact with MFLTS S2S:

- Smartphone only: The two text fields in Figure 3b also serve as buttons that, when pressed, start recording speech from the phone's internal microphone. Upon lifting the finger the translation starts.
- Peripheral: As seen in Figures 3a and 4, we created a peripheral microphone device that, alongside a battery-driven speaker, greatly enhances interaction.

Figure 4: The application tested during a live exercise (AEWE)

1.3 Evaluation

When building a complex application such as an S2S application, it is not immediately obvious how to evaluate it in order to make improvements. The simplest way, and this was done in the early days of the TransTac program, is by evaluating the systems through their individual components' performance:

- Speech recognition: Word Error Rate (WER)
- Machine translation: BLEU (and others)

Because WER and BLEU scores are easily generated and compared, they are instinctively chosen for evaluation, but there is an assumption riding on using these low-level statistics, which is that an improvement in either of those scores translates to an improvement in the usability of the application.

What we found during the many iterations of the application is that, often that is not the case. In fact, minute changes in the user interface often would cause far more drastic improvements in user satisfaction. Because of this, later evaluations added the measurement of "High Level Concept Transfer": During an evaluation a soldier would be given a list of specific information he is trying to establish (e.g., "what days of the week do supply trucks come through this town?") by asking the FLE. Systems were then compared by how many concepts (i.e. pieces of pertinent information) they were able to transmit in a given time period. By the time of the MFLTS program, the software was being evaluated in mock exercises where it was being used (successfully) to gather actionable intelligence that helped soldiers achieve their mission objective.

1.4 The Right Flow

A key realization during the development was, for lack of a better description, the system's "place in the conversation". Given such a powerful system, it is tempting to elevate its interactions to the level of a human interpreter by having it inject itself into the conversation like an interpreter would. As an example, an interpreter might ask for clarification from the soldier ("did you mean 'magazine' like the warehouse, or the gun magazine?"). This type of interjection was in fact tested in more detail in a subsequent research project (DARPA *BOLT*), but the problem that arises with this approach is that it essentially adds a third party into the conversation. Not only that, it also consumes precious time that the two parties involved have to wait.

As a result, we followed the following guideline during design: **Any system output comes at a premium**. This was a very consistent trend, as shown in three different aspects of the application:

- English confirmation: The idea was to use English TTS to confirm what the system's ASR had recognized, with the expectation that the soldier would interrupt the system if the ASR was wrong. We found that soldiers rather preferred to deal with any ASR error during follow-up conversation than restarting the utterance. The flow of the interaction was more important than this additional checkpoint.
- Backtranslation: Similar to English confirmation, but instead the foreign-language text was again back-translated to English, and then put out with TTS before the foreign TTS was played. Once again, the slowdown of the conversation was deemed too onerous over the additional piece of information.
- Abort: Innocuous as it may seem, the ability to quickly abort an ongoing translation drastically improved usability (initially the soldier had to wait out the translation before beginning a new one).

The key conclusion here was that both parties want the system "out of the way" of the conversation. That is, in a heavily multimodal human conversation (facial expressions, body language etc.) the system needs to facilitate information flow, not take it over or manipulate it. In terms of Enfield's "conversation engine" (Enfield, 2018), the goal is to maintain the flow of said engine as much as possible.

1.5 The Right Hardware

In a similar vein to the previous section, finding the right physical representation of the device can either facilitate or hinder the interaction. For example, one of the devices that was tested but ultimately rejected was a telephone receiver that allowed the FLE to listen to the translation and speak into it, to piggy-back on the familiarity of people with a telephone conversation. However, the necessary physical proximity to the soldier with this device was rated too uncomfortable to both parties, as were the possible social implications of a foreigner "receiving a phone call" from a US soldier.

Instead, we opted to emulate another interaction most people are familiar with: a TV interview, where one person with a microphone interviews the other. Several aspects made this way of interacting stand out:

- All devices are in possession of the soldier, and he/she can decide how far or close the microphone is to the FLE
- The act of physically pointing the microphone either at the soldier's mouth or at the FLE's mouth is an implicit queue of "it is your/my turn to speak". As mentioned above, this type of non-verbal communication is almost always preferrable to voice-based queueing.

1.6 The Right Person

A different, interesting realization was that the success of the application is as dependent on its software as it is on the person that uses it. We have consistently experienced vast ranges of user reports, from ”this app hindered my attempts to communicate” to ”this was almost like I speak the language myself”. What we found is that in any given group, there will be people naturally predisposed to using the app: for these people, and it is astonishing to witness, the application becomes second nature, and they return to focusing on observing the FLE speaking, as if they were conversing in their own native language. Just as with any other tool, success of the application comes down to selecting a ”Communicator” in the group who shows natural adeptness.

That said, the MFLTS program requires basic proficiency of the app by any soldier within one hour, which we achieved by an interactive training embedded in the Android application. Leveraging soldiers' innate familiarity with smartphone user interfaces, usually it is rather a matter of minutes after which they then start focusing on mastering the social subtleties of the application.

2 Conclusion

The MFLTS S2S application has shown its value as a translation application under real-life constraints, and development is ongoing. There are many avenues that should be explored:

- Hands-free: To return even further to the ideal of an unimpeded conversation, the system would not have to be told when, or who, is speaking at a moment. It should detect the spoken language, and present its translation at an opportune time.
- Even smaller: Powerful smartphones are ubiqitous, but their different usage profile demand a size that is not necessarily needed for a translation device. At the same time, single-board computers (SBCs) are upcoming that could be used to create even more integrated devices.
- Far more advanced: A major incurrence of conversation latency is the current requirement to wait until the person has stopped speaking before the translation can be spoken out, simply because there would otherwise be two people speaking. An advanced approach such as directional speakers might allow for truly real-time translation where partial translations are output while the person is still speaking. An exciting array of considerations (translation accuracy vs latency etc) arise from this.

As mentioned before, what must be used as the ultimate goal is the ”absence” of the tool, i.e., the application. Humans are masters at conversations, and any translation application needs to strive to return to that realm.

References

David Stallard, R. P. e. a. (2011). The BBN TransTalk Speech-to-Speech Translation System. In *Speech and Language Technologies*, chapter 3. InTech.

Enfield, N. (2018). *How We Talk: The Inner Workings of Conversation.*

Meermeier, R. and Colbath, S. (2017). Applications of the BBN Sage Speech Processing Platform. In *Proceedings of Interspeech 2017*, Stockholm, Sweden.

Roger Hsiao, R. M. e. a. (2016). Sage: The New BBN Speech Processing Platform. In *Proceedings of Interspeech 2016*, San Francisco, USA.