Design Issues for a Bidirectional Mobile Medical Speech Translator

Nikos Tsourakis
ISSCO/TIM/ETI
University of Geneva
Switzerland
Nikolaos.Tsourakis@unige.ch

Pierrette Bouillon
ISSCO/TIM/ETI
University of Geneva
Switzerland
Pierrette.Bouillon@unige.ch

Manny Rayner
ISSCO/TIM/ETI
University of Geneva
Switzerland
Emmanuel.Rayner@unige.ch

ABSTRACT
We argue that there is an urgent need for spoken dialogue translation systems in the medical domain. In this work we describe how an existing system of this kind, originally intended for a desktop PC, was adapted to run in a mobile environment. The different characteristics of the target device necessitate new interaction approaches and interface design. The new configuration uses two client applications running on two different devices, one for the doctor and one for the patient, together with a server; the components are connected over a wireless network. The system supports context-dependent translation in both directions. We give a general overview of the system, and discuss some of the relevant design issues pertaining to deployment of speech-enabled systems on mobile devices.

Categories and Subject Descriptors
H.5.2 **[Information Interfaces and Presentation]**: User Interfaces – *graphical user interfaces (GUI), natural language, voice I/O*

General Terms
Design, Human Factors

Keywords
Medical Speech Translation, MedSLT, Regulus

1. INTRODUCTION
The world's current population of 6.6 billion speaks more than 6,000 languages [1]. Language barriers often cause inconvenience. When medical issues are involved, however, they cease to be mere inconvenience, and can become life-threatening. Anyone who has had the misfortune to fall ill in a country where they do not speak the local language will be painfully aware of this from their own experience, and quantitative studies [2, 3] have shown that lack of a common doctor-patient language correlates with a greatly increased probability of negative outcomes across a variety of objective indicators. Despite this fact, in the United States, where 52 million people speak a language other than English at home and 23 million people have limited English proficiency (LEP) [4], one study found that about half of LEP patients presenting to an emergency department were not provided with a medical interpreter [5]. Unfortunately, trained medical translators are both scarce and expensive. The substantial gap between the need for and availability of language services in health care could be bridged through effective medical speech translation systems, e.g. [6], [7], [8].

MobileHCI 2009, September 15–18, 2009, Bonn, Germany.

MedSLT system [6] is a multilingual spoken language translation system tailored for medical domains. The system is designed to help in situations where no common language exists between the doctor performing the diagnosis and the patient. MedSLT is based on Regulus [9], an Open Source platform that supports construction of rule-based medium-vocabulary multilingual spoken dialogue applications. MedSLT has already been deployed in different versions on a desktop PC and several system and positive user evaluations have been reported in earlier publications [10, 11, 12, 13, 14].

At the international workshop on medical speech translation held in conjunction with the 2006 NAACL conference (http://www.sehda.com/hlt-2006-workshop/), doctors and other potential users several times said that a system like MedSLT would be far more useful to them if it was available on a hand-held device, rather than, as with the present version, on a laptop. Systems like [15], [16], [17] are efforts towards the deployment of mobile speech-to-speech translation applications.

The paper describes a port of the MedSLT system to this type of environment. Rather than trying to put all the functionality on the mobile devices, which is still very challenging, we use the distributed client-server architecture described in [18], in which a centralized server accommodates the burden of executing resource-hungry processes (in particular, most of the recognition task and the natural language processing), and the load on the client becomes light enough that it can easily be hosted on a mobile device.

However the limited screen display on the client platform, in addition to the different input mechanisms compared to a desktop PC, necessitates alternative interaction approaches. Integration of the different functionalities to provide efficient communication is the main challenge. In this work we try to address some of the issues caused by the special nature of the interaction between a novice user (the patient) and an expert user (the doctor), and in particular provide a different GUI for each of the two parties.

The paper is organized as follows. In Section 2 we give a short overview of the Regulus platform. In Section 3 we present the desktop version of the system along with the features embedded in the mobile version. We continue in Section 4 with some issues concerning the deployment on a mobile platform. A description of the system's architecture is presented in Section 5. In Section 6 we focus on the GUIs for both the doctor and the patient. The results of a short evaluation are presented in Section 7. The final section concludes.

2. REGULUS PLATFORM

Regulus is an Open Source platform that supports construction of rule-based medium-vocabulary spoken dialogue applications. As well as MedSLT, it has already been used to build several other substantial speech-enabled applications. The most prominent of these is NASA's Clarissa procedure navigator [19], which reached to the point of initial testing on the International Space Station[1].

The distinguishing feature of the Regulus system is the emphasis on principled use of linguistically motivated methods; all speech and language processing is performed using resources ultimately derived from substantial, domain-independent feature grammars, suitably compiled for the tasks of analysis, generation and speech recognition. Early versions of the platform used one base grammar per language; more recently, we have even proceeded beyond this point, and merged together the resource grammars for related languages [20]. Compilation of the general grammar into its final form proceeds in several stages, and involves example-based methods, driven by small corpora, which make it possible to transform the loose general grammar into tightly constrained domain-specific grammars. For the case of performing speech recognition, subsequent processing compiles the domain-specific grammar into a Grammar-Based Language Model (GLM) in Nuance format.

The Regulus platform also contains further infrastructure to support construction of applications which use the recognizers, parsers and generators as components. In both cases, the main processing flow consists of a pipeline. Thus processing in a speech translation application starts with speech recognition (including parsing), which produces a source language semantic representation. This representation is then passed to a translation engine, which is interlingua-based. It first translates the source representation into an interlingua form, and then into a target language representation. Finally, the target language grammar, compiled into generation form, is used to create a target language surface string.

The generic dialogue application architecture is similar. The central component is the Dialogue Manager (DM), which receives dialogue moves and produces abstract actions. It also manipulates an information state, which maintains context; processing will generally be context-dependent. The DM is bracketed between two other components, the Input Manager (IM) and the Output Manager (OM). The IM receives logical forms and non-speech inputs if there are any, and turns them into dialogue moves. The OM received abstract actions and turns them into concrete actions. Usually, these actions will be either speaking, though text-to-spech or recorded speech, or manipulation of a GUI's screen area.

[1] http://ti.arc.nasa.gov/project/clarissa/results/

3. DESKTOP VERSION

The bidirectional MedSLT system supports two configurations. The first of these is a restricted version allowing yes/no and WH-questions on the doctor side, and elliptical (short) answers on the patient side. The second is a less restricted version, which in addition supports patient answers in the form of full sentences. In the initial prototype, the doctor language is English, French, Japanese, Arabic or Catalan, and the patient language is Spanish. The following table shows examples of grammar coverage for the English language:

Table 1. MedSLT Grammar Coverage Examples

Where? Is the pain above your eye?	**When?** Have you had the pain for more than a month?
How? Is it a stabbing pain?	**How often?** Do you get headaches several times a week?
How long? Does the pain typically last a few minutes?	**Associated symptoms?** Do you vomit when you get the headaches?
Why? Does bright light make the pain worse?	**What helps?** Does sleep make the pain better?

Figure 1 shows a screenshot of the GUI for the desktop version of the system. It contains two tabs, one for the doctor and one for the patient, with the two parties normally taking alternate turns. The interaction in this version proceeds as follows. Initially the doctor asks a medical diagnosis question. The system produces two separate results: a back-translation (translation from English-to-English) that shows how the system understood the question, and a list of similar "help examples", based on ones taken from the development corpus, which are already known to work correctly. The back-translation and help examples are discussed immediately below in Sections 3.1 and 3.2.

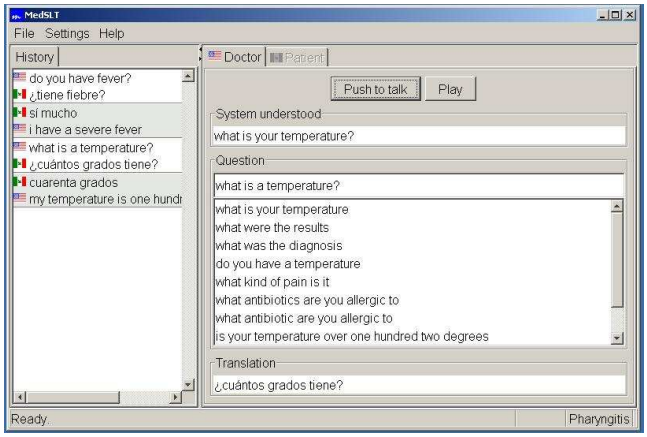

Figure 1. Desktop bidirectional GUI

The user either approves the system's back-translation, or else selects a help example; in either case, the selected utterance is passed to the translation module, and the resulting translation is presented in the "Translation" box at the bottom of the screen. When the user presses the Play button, the system speaks the target utterance aloud, and then transfers control to the other partner in the dialogue. The history of the dialogue is displayed in the left-hand pane.

3.1 Help system

In a spoken dialogue application, it is often desirable to give users immediate feedback on the system's coverage, which is particularly likely to be useful when recognition has been unsuccessful. Work by [21] suggests that most users are able to use this kind of feedback to gain rapid familiarity with the system's coverage.

The help module for a Regulus-based application works in the following way. The system is provided with a library of examples, compiled during system development, which are known to be within coverage. At runtime, the application carries out a second round of recognition using a backup recognizer equipped with a Statistical Language Model (SLM), and uses a surface N-gram based distance metric to order the library examples by their closeness to the SLM recognizer's output; the top few examples are then shown to the user. This happens independently of whether the user's original input was correctly recognized or not, which means that for each input, the user always sees a set of related in-coverage phrases.

The experiments described in [10, 22] confirm our intuitive observation that the help module makes a large difference to usability. With help available, most new users rapidly acquire confidence; without it, they flounder. Table 2 shows some help examples for the input sentence: "Have you had a strep test?".

Table 2. Help System Example

User	*"Have you had a strep test?"*
System	Was a strep test carried out? Was a rapid strep test done? How long have you had congestion? How long have you had a headache? What where the results of a strep test?

3.2 Back-translation

Back-translation is a way of rephrasing the user input and is an indication of what was actually understood by the system. This is particularly important in cases where the interpretation differs significantly from the surface input, for example when it consists of an incomplete utterance like a short answer, which needs to be resolved with reference to the preceding dialogue context. In order to produce the back-translation, we first translate the recognition result into interlingua form, and then translate it back the source language. Given that the system is already capable of multi-lingual translation, and in particular of realizing an interlingua form in the different languages, this strategy is very easy to realize. Table 3, depicts some examples showing how the back-translation differs from the actual user's input:

Table 3. Back-translation Examples

User	Back-translation
"Does red wine cause any headaches?"	Do you have the headaches when you drink red wine?
"What relieves your pain?"	What makes the pain better?
"How about bright light?"	Is your headache made worse by bright light?
"Do you have it every day?"	Does the pain occur every day?

4. ISSUES WITH THE MOBILE VERSION

In this section we will address some specific issues related to the mobile version of the system. Initially, we should mention that this version is less ambitious in terms of supporting patient feedback, which so far can only be non-verbal.

As we mentioned in the introduction we have two kinds of users, with different levels of expertise. On the one hand, we have the doctors, it is reasonable to assume that they will use the system regularly, and will be able to learn to use it quickly at an expert level. On the other hand, we have the patient, who may have to interact with the system only once in his life. This variation in users, demands different kind of interaction style for each party. As discussed in the following sections, we have designed two different GUIs. The one for the doctor is more sophisticated, although still fairly simple for a regular user. The one for the patient supports no speech recognition with the user selecting the desired response through a touch-screen.

Portability is another important aspect. Consider, for example, a doctor making ward rounds. It would be much comfortable to carry two small devices instead of a laptop. On the other hand, it is not always feasible for a patient to sit side by side with the physician in front of a computer. If, as is commonly the case, he is lying on a bed, it would be much easier to interact with a mobile device. Moreover the clinical nature of the patient's problem may demand minimal interaction (e.g. sore throat etc).

The distributed nature of our system offers the flexibility to put all the resource hungry processes in powerful centralized servers. Additionally we can easily perform changes in these servers without the need of downloading and installing a new application on the target device. A change in a recognition grammar or the help mechanism will be invisible to the end user, but will enhance the performance of the system.

The decentralization of tasks has also another benefit, as it reduces the CPU load and consecutively power consumption. This would be an important factor for a professional who doesn't need to charge the device frequently.

Of course the deployment of a mobile device has other drawbacks. For example the small display prohibits the presentation of all the information in separate text boxes. In our approach we merged the back-translation output and the help examples in just one list. For the user all items in the list represent possible options. Again the history of the dialogue is hidden and can be presented upon request. On the patient's side the GUI is minimal, with very few big buttons and text in large fonts.

5. SYSTEM ARCHITECTURE

Figure 2 shows the top-level components of the network. The system uses a distributed architecture, where the various nodes are configured as autonomous peers in the same network, and offer different kinds of services.

The mobile device, which is the only part the user sees, contains all the logic needed to communicate with the other peers. In our configuration there is one device for each of the interacting parties When the user speaks to the device, audio packets are transmitted through the wireless network to the ASR server, where they are recognized using both the grammar-based and the statistical recognizers. The recognition results, in the form of N-best speech hypothesis lists, are sent to the translation server. This performs

all necessary natural language processing; its output is the translation response, together with a set of help sentences. The two mobile applications (at present, deployed on Nokia N810 Internet Tablets) exchange information when one of the participants confirms a response.

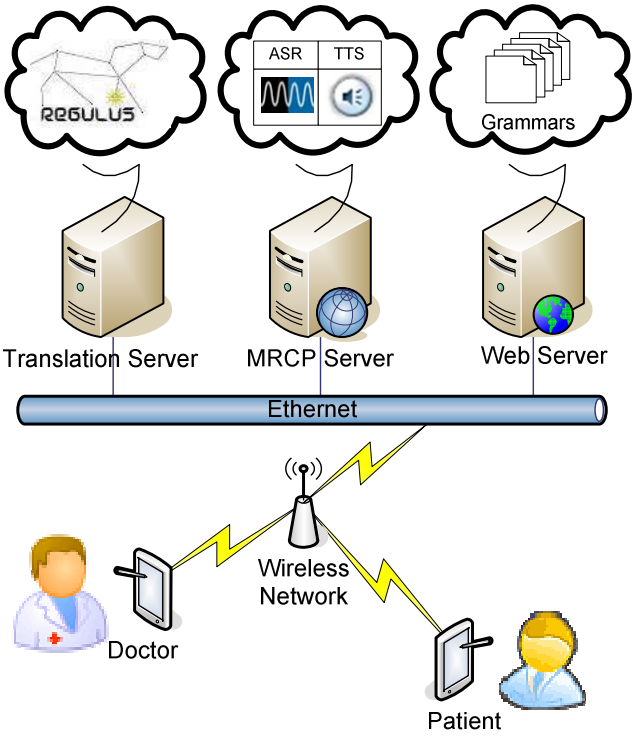

Figure 2. Mobile System Architecture

6. MOBILE APPLICATION

As mentioned earlier, the different levels of expertise and the different physical condition of the doctor and the patient require two different graphical user interfaces. In the two next subsections we will give a short overview of the features for both applications.

6.1 Doctor's GUI

The doctor's mobile application is a lightweight process, implemented in C++, responsible for the following tasks:

- Supporting different input modalities (speech, pen buttons) and different output modalities (screen and speakers).
- Communicating and requesting services from the Media Resource Control Protocol (MRCP [23]) server for either ASR or TTS.
- Capturing and packetizing streaming audio (8KHz - 8bit) using the Real-Time Transport Protocol (RTP over UDP).
- Forwarding the recognition result to the Translation Server in order to perform the natural language processing.
- Providing the answer to a user's request and presenting a set of help sentences according to the user's input.

As seen in Figure 3, the doctor side GUI consists of two text boxes for presenting:

- A list with the back-translation (grammar-based recognition) followed by the help examples associated with the statistical recognition.
- The translation of the selected output.

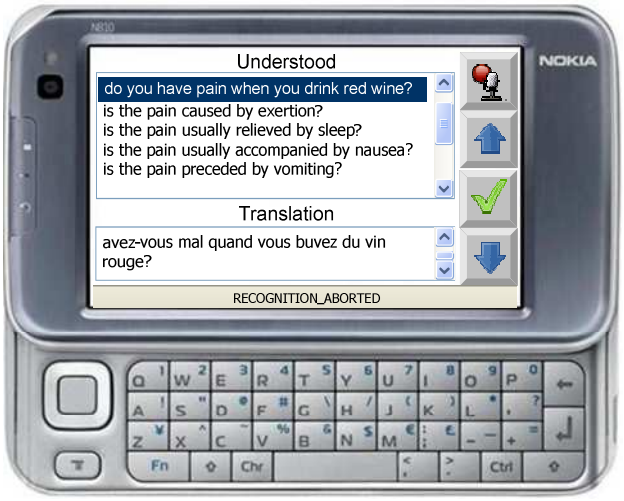

Figure 3. Doctor's GUI

There is also a button for initiating the speech recognition and three buttons for navigating in the result list. The same functionalities are also offered by the hardware buttons located in the upper left side of the device. Finally, the status bar presents different events associated with the user's interaction (e.g. start of speech, recognition completed etc.).

The doctor can navigate through the list and pick the response that he prefers. When the tick button is pressed the selected item is translated to the patient's language and it is presented and announced on the latter's device. By pressing a specific hardware button the physician can also observe the dialogue history between the two parties.

6.2 Patient side GUI

The GUI for the patient is much simpler. As we can observe in Figure 4, it consists of a textbox for presenting the doctor's question and six buttons for the most useful responses, which cover the basic subset of the possible answers (in French).

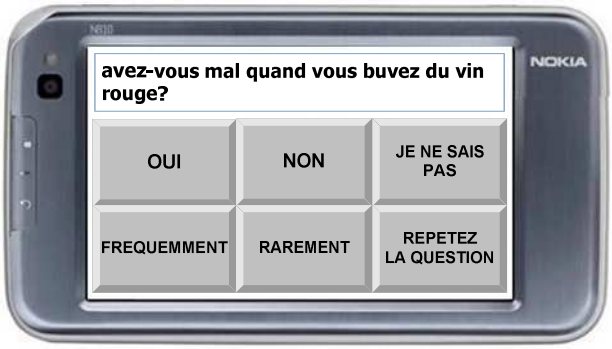

Figure 4. Patient's GUI

The patient's responses are categorized into three group pairs, so that buttons answering the same type of questions are in a close position. These include:

- A Yes/No response, which can be used in questions like "Do you have fever", "Does the pain appear every morning" etc.
- A Frequent/Seldom response, for questions like "Does bright light make your headaches worse", "Is it a persistent pain" etc.
- A Don't know/Repeat response, when the user doesn't know what to answer or he is not able to clearly read the text on the screen and wants to listen to the audio output again.

The user can tap with his finger on one of these buttons, which results in the response being announced back on the doctor's device, in his language. In the next version of the GUI, we will also include forms where the patient can specify additional information e.g. fever temperature, date intervals, medications list etc.

The following table shows a sample interaction between the doctor and the patient:

Table 4. Sample Interaction

Doctor	Input	*"Is the pain stabbing?"*
	Translation	*La douleur est-elle comme un coup de poignard?*
Patient	Press	**NO**
Doctor	Input	*"Throbbing?"*
	Translation	*La douleur est-elle pulsative?*
Patient	Press	**I DON'T KNOW**
Doctor	Input	*"Does the pain come early in the morning?"*
	Translation	*La douleur survient-elle tôt le matin?*
Patient	Press	**YES**

7. EVALUATION

We evaluated the application with respect to speech recognition performance and response latency, and obtained results consistent with those of our previous studies [18]. The fact that we use a distributed client-server architecture implies that the mobile application can benefit from its ability to run resource-hungry processes on the remote peer. The issue we wished to investigate, however, is whether this also involves degradation on the quality of the system's understanding of the user's input.

For our experiments we used the data collected by six non-native English speakers in an office environment. Each speakers had to read 50 selected, in coverage sentences during four interaction scenarios. We thus collected 200 sentences from each user, producing a total of 1200 waveforms (300 waveforms per interaction scenario). The scenarios were the following:

1. The user speaks to the desktop PC using a simple office headset (DES_H).
2. The user speaks to the mobile device using the same headset as the one used for the desktop PC (MOB_H).
3. The user speaks to the mobile device using the onboard microphone from a distance (MOB_O).
4. The user speaks to the system using the Nokia BH-900 Bluetooth headset with a sliding boom microphone (MOB_B).

The error rates for each interaction scenario are presented in Table 5. We present figures for three metrics: Word Error Rate (WER), Sentence Error Rate (SER) and Semantic Error Rate (SemER).

Table 5. Error Rates per Interaction Scenario

	Desktop DES_H	**Mobile MOB_H**	**Mobile MOB_O**	**Mobile MOB_B**
WER	**6.7%**	**6.3%**	**12.8%**	**10.2%**
SER	**29.7%**	**29.7%**	**37.0%**	**35.3%**
SemER	**11.0%**	**11.3%**	**23.0%**	**18.3%**

SER is, as usual, defined as the proportion of utterances where at least one word is misrecognized. Semantic Error Rate is defined as the proportion of utterances which produce a semantic representation, at the level of dialogue processing, which is different from the one which would have been produced given perfect recognition. SemER can thus be thought of as a version of SER that has been adjusted to take into account the fact that many recognition errors have no effect on system response.

From the results presented earlier, we can observe similar performance when using the headset on the desktop PC and the mobile device. This was more or less an expected result. Besides any hardware differences between the two platforms, the factor that mainly differentiates them and may affect the performance is the wireless data network. As our architecture is distributed, we rely on the underlying data network mainly for audio transmission. Audio is always time sensitive information and a congested network will cause packet loss (transmission over the UDP connectionless protocol). In our experiments, we used the public wireless network of the University of Geneva, which offered a reliable and speedy access medium.

In the case of recording with the onboard microphone, we observe a clear degradation in the performance. From our observations on the corresponding waveforms we see that the distance definitely affects the quality of the speech. One may argue that the user can bring close to his mouth the device when needed. This usually causes clipping on the waveforms as the user speaks to the device too close. On the other hand a constant movement of the device may affect the smooth interaction between the user and the system.

The Bluetooth headset is a compromise, as it provides a performance between the two extremes and on the other hand it offers the flexibility of no cables. We believe that for less demanding applications we could rely on the reliability of this headset.

Besides the previous metrics concerning the speech recognition performance we calculated the response latency. This is defined as the time between the press of the recognition button and the

acquisition of the recognition result (including the time of user's speech). As seen in Table 6, we have an overhead of 0.63 seconds in the mobile version compared to the desktop one. We believe that this is a minimal compromise considering the benefits of a possible transition to the mobile version of our system.

Table 6. Recognition Response Latency per Scenario

	Desktop DES_H	Mobile MOB_B
Time (sec)	**2.838**	**3.470**

Another interesting comparison is between the latency of acquiring the translation result on both versions. This is defined as the elapsed time after sending the request on the translation server and obtaining the output. Table 7 shows the results of this comparison, which again indicate a negligible latency for real time applications.

Table 7. Translation Response Latency per Scenario

	Desktop DES_H	Mobile MOB_B
Time (sec)	**0.479**	**0.634**

8. CONCLUSIONS & FUTURE WORK

In this work, we tried to identify some key issues when moving from the desktop version of our MedSLT system to the mobile one. We first examined the main features embedded on both versions. By incorporating client/server architecture we offer flexibility and scaling on the target application. The different user interaction styles demand specific solutions for each of the engaging parties. Finally, we showed that the mobile version has a similar performance concerning the speech recognition as long as the same headset is used.

In the next phase of this work, we will perform experiments with real users. We will use medical students as we did in previous studies, together with users that impersonate the patients. The participants will simulate different, well-scripted, fixed scenarios. In these experiments a task will be considered as successfully accomplished when a doctor correctly identifies the type of medical condition that the patient was asked to simulate. One important aspect is also the time needed for a physician to pick the correct diagnosis from the list of candidates. Recognition and translation errors along with interaction errors should be quantified for both interacting parties. The same work already implemented for the desktop version will offer a comparison basis.

Considering the fact that the participants may interact under stress, or in a noisy environment, we could offer additional modalities. For example a question can be picked from a predefined list or written with the stylus pen. In any case the question of synergies between modalities is a topic closely related with the current work.

A more ambitious task would be to add meta-linguistic information in the generation grammars. Using SSML tags for the TTS output we can give emphasis to certain parts, change the pitch etc.

Another way to deliver meta-linguistic information could be the addition of images. For example when the doctor asks "Did you eat salmon yesterday?" and the user doesn't know what kind of fish is that he could press on the corresponding icon. For modeling certain kind of pain, we can include voice icons that resemble the pain.

A model of the human body on the screen could also be appropriate for depicting problematic areas to the doctor. Other types of iconic representations, such as temperature scales, date intervals, and medications list are also possible.

9. ACKNOWLEDGMENTS

We would like to thank Nokia and its University Donation Program for offering two N810 Internet Tablets for use in the current work.

10. REFERENCES

[1] Graddol, D. 2004. The future of language. Science, 303:1329–1331.

[2] Flores, G. 2005. The impact of medical interpreter services on the quality of health care: A systematic review. Medical Care Research and Review, 62:255–299.

[3] Flores, G. 2006. Language barriers to health care in the United States. New England Journal of Medicine, 355:229 231.

[4] U.S. Census. 2007. Selected Social Characteristics in the United States: 2005. Data Set: 2005 American Community Survey.

[5] Baker, D.W., Parker, R.M., Williams, M.V., Coates, W.C., and Pitkin, K. 1996. Use and effectiveness of interpreters in an emergency department. Journal of the American Medical Association, 275:783–8.

[6] Bouillon, P., Flores, G., Starlander, M., Chatzichrisafis, N., Santaholma, M., Tsourakis, N., Rayner, M. and Hockey, B. A. 2007a. A Bidirectional Grammar-Based Medical Speech Translator. In Proceedings of the ACL Workshop on Grammar-Based Approaches to Spoken Language Processing, Prague Czech Republic.

[7] Precoda K., Zheng J., Vergyri D., Franco H., Richey C., Kathol A., and Kajarekar S. 2007. "IraqComm: a next generation translation system". Proceedings of Interspeech 2007, Antwerp, Belgium.

[8] Fluential S-MINDS speech-to-speech translation system, http://www.fluentialinc.com.

[9] Rayner, M., Hockey, B.A. and Bouillon, P. 2006. Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler. Stanford, California, CSLI Press.

[10] Chatzichrisafis, N., P. Bouillon, M. Rayner, M. Santaholma, M. Starlander, and B.A. Hockey. 2006. Evaluating task performance for a unidirectional controlled language medical speech translation system. In Proceedings of the HLT-NAACL International Workshop on Medical Speech Translation, pages 9–16, New York.

[11] Bouillon, P., Flores, G., Georgescul, M., Halimi, S., Hockey, B.A., Isahara, H., Kanzaki, K., Nakao, Y., Rayner, M., Santaholma, M., Starlander, M. and Tsourakis, N. 2008.

Many-to-Many Multilingual Medical Speech Translation on a PDA. Proceedings of AMTA, Waikiki, Hawai'i,

[12] Rayner, M., Bouillon, P., Brotanek, J., Flores, G., Halimi, S., Hockey, B.A., Isahara, H., Kanzaki K., Kron, E., Nakao, Y., Santaholma, M., Starlander, M. and Tsourakis, N. The MEDSLT 2008 system. 2008. Proceedings of Workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, Coling 2008, Manchester, England.

[13] Starlander M., Bouillon, P., Flores, G., Rayner, M. and Tsourakis, N. 2008. Comparing Two Different Bidirectional Versions of Limited-domain Medical Spoken Language Translator MedSLT, Proceedings of EAMT, Hamburg, Germany.

[14] Starlander, M. and Estrella, P. 2009. Relating recognition, translation and usability of two different versions of MedSLT. To appear in the MT Summit, Ottawa, Ontario Canada.

[15] Zhang Y. and Vogel S. 2007. PanDoRA: a large-scale two-way statistical machine translation system for hand-held devices. Proceedings of MT Summit, Copenhagen, Denmark.

[16] Waibel A., Badran A., Black A.W., Frederking R., et al. 2003. “Speechalator: two-way speech-to-speech translation on a consumer PDA”. Proceedings of Interspeech 2003, Geneva, Switzerland.

[17] Voxtec Phraselator mobile translation system, http://www.voxtec.com/.

[18] Tsourakis, N., Georgescul, M., Bouillon, P. and Rayner, M. 2008. Building Mobile Spoken Dialogue Applications Using Regulus. Proceedings of the LREC 2008, Marrakech, Morocco.

[19] Rayner, M., Hockey, B.A., Chatzichrisafis N., Farrell, K. and Renders, J.-M.. 2005. A voice enabled procedure browser for the International Space Station. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, Michigan.

[20] Bouillon P., Rayner M., Novellas, Vall, B., Starlander, M., Santaholma, M., Nakao, Y. and Chatzichrisafis, N. 2007. Une grammaire partagée multi-tâche pour le traitement de la parole : application aux langues romanes, Traitement Automatique des Langues, Volume 47, 3/2006, Hermes & Lavoisier.

[21] Hockey, B.A., Lemon, O., Campana, E., Hiatt, L., Aist, G., Hieronymus, J., Gruenstein, A., and Dowding, J. 2003. Targeted help for spoken dialogue systems: Intelligent feedback improves naive user’s performance. In Proceedings of the 10th EACL, Budapest, Hungary.

[22] Rayner, M., Bouillon, P., Chatzichrisafis, N., Hockey, B.A., Santaholma, M., Starlander, M., Isahara, H., Kanzaki, K., and Nakao, Y. 2005. A methodology for comparing grammar-based and robust approaches to speech understanding. In Proceedings of the 9th International Conference on Spoken Language Processing (ICSLP), Lisbon, Portugal.

[23] Shanmugham, S., Monaco, P. and Eberman, B. 2005. A Media Resource Control Protocol (MRCP) Developed by Cisco, Nuance, and Speechworks. Internet Engineering Task Force.