Building Mobile Spoken Dialogue Applications Using Regulus

Nikos Tsourakis, Maria Georgescul, Pierrette Bouillon and Manny Rayner
ISSCO/TIM/ETI University of Geneva
Boulevard du Pont-d'Arve, CH-1211 Genève 4, Switzerland
E-mail: Nikolaos.Tsourakis@issco.unige.ch, Maria.Georgescul@eti.unige.ch, Pierrette.Bouillon@issco.unige.ch, Emmanuel.Rayner@issco.unige.ch

Abstract

Regulus is an Open Source platform that supports construction of rule-based medium-vocabulary spoken dialogue applications. It has already been used to build several substantial speech-enabled applications, including NASA's Clarissa procedure navigator and Geneva University's MedSLT medical speech translator. System like these would be far more useful if they were available on a hand-held device, rather than, as with the present version, on a laptop. In this paper we describe the Open Source framework we have developed, which makes it possible to run Regulus applications on generally available mobile devices, using a distributed client-server architecture that offers transparent and reliable integration with different types of ASR systems. We describe the architecture, an implemented calendar application prototype hosted on a mobile device, and an evaluation. The evaluation shows that performance on the mobile device is as good as performance on a normal desktop PC.

1. Overview

Regulus (Rayner et al, 2006) is an Open Source platform that supports construction of rule-based medium-vocabulary spoken dialogue applications. The distinguishing feature of the 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 (Bouillon et al 2007). 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 recognition, subsequent processing compiles the domain-specific grammar into a Grammar-Based Language Model (GLM) in Nuance format.

An important component of the overall Regulus approach is that applications in general include an integrated help system, whose purpose is to alleviate the lack of robustness inherent in a purely rule-based recognition architecture. After each utterance, the help system provides the user with in-coverage examples, chosen to be as close to the user's actual utterance as possible. Our experience is that most users are able to use this kind of feedback to gain rapid familiarity with the grammar's coverage. The help module's output is based on a precompiled library of utterances, which have already been evaluated during system regression testing as being within grammar coverage and producing correct responses. At runtime, the application carries out a second round of recognition using a backup recognizer equipped with a Statistical Language Model (SLM); it passes the result to the dialogue server, which returns a list of examples from the library which is similar to the statistical recognizer's result. Similarity is currently computed in terms of a backed-off surface N-gram metric (Starlander et al 2005).

Regulus has already been used to build several substantial speech-enabled applications, of which the most prominent are NASA's Clarissa procedure navigator (Rayner et al 2005) and Geneva University's MedSLT medical speech translator (Bouillon et al 2005). Performance is at a level where it is very reasonable to think of using systems like these in real-world situations. Clarissa reached the point of initial testing on the International Space Station[1]; MedSLT has been successfully used by medical students with no previous exposure to the system to perform diagnosis tasks on simulated patients. In particular, they were able to learn the coverage of the system entirely by using a help system of the kind described above (Chatzichrisafis et al, 2006).

Although these results are promising, they bring to the forefront another important consideration: speech enabled systems need to be deployed on easily portable platforms if they are to realize their full potential. At the international workshop on medical speech translation (Bouillon et al 2006), emergency doctors and other potential users several times said that a system like MedSLT would be far more useful to them if it were available on a hand-held device, rather than, as with the present version, on a laptop. Similarly, one of the most frequent comments the Clarissa team received from NASA astronauts was that the system ideally should run on a hardware platform which could be taken into cramped or enclosed spaces.

This paper describes the Open Source framework we have developed over the last year, which makes it possible to run Regulus applications on generally available mobile platforms with performance essentially no worse than on a desktop machine. Although it is feasible to put medium-vocabulary systems like the Regulus

[1] http://ic.arc.nasa.gov/projects/clarissa/iss_report.html

applications described above on a PDA (cf. for example (Waibel et al 2003)), performance is significantly worse than on a desktop, and most standard recognition software will not run in the PDA environment. Use of the statistical recognizers needed for the help system is particularly problematic. For these reasons, we have chosen to implement a distributed client-server architecture. Centralized servers can accommodate the burden of executing resource-hungry processes (in particular, most of the recognition task), and the load on the client becomes light enough that it can be hosted on a mobile phone. Our solution is closely modeled on that implemented in (Tsourakis et al 2006), though we have adapted the architecture to make full use of MRCP (Shanmugham et al 2005), a protocol stack proposed by W3C for managing ASR and TTS engines over a distributed network. This particular mechanism offers transparent and reliable integration with different types of commercially available ASR systems, including in particular Nuance, in the form of an easily extensible generic infrastructure.

Section 2 describes our architecture in more detail. Section 3 describes a speech-enabled calendar application, available from the Regulus website, which is intended to serve as a paradigm example of how to implement an application of this type. Section 4 describes the evaluation steps of our work. We will use the calendar application, run on a Nokia Linux N800 Internet Tablet, to demonstrate the general mobile application framework.

2. System Architecture

As already mentioned, the system uses a distributed architecture. 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.

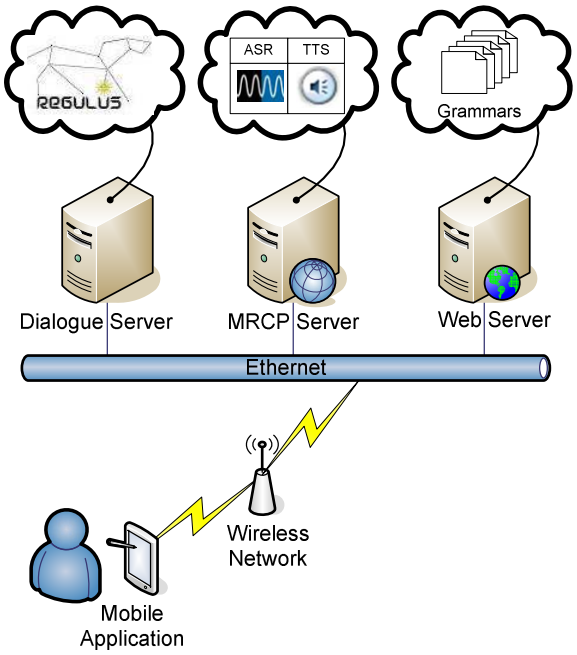

Figure 1: Network components for the mobile application

Figure 1 shows the top-level components of the network. When the user speaks to the device, audio packets are transmitted through the wireless network to the ASR server, where it is 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 Dialogue server. This performs all necessary natural language processing; its output is the dialogue response, together with a set of help sentences

2.1 Mobile Application

The 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 MRCP server.
- Capturing and packetizing the audio (8kHz,8bit).
- Forwarding the recognition result to the Dialogue 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.

2.2 Dialogue Server

The dialogue server is implemented on top of the Regulus platform, and is responsible for all natural language processing. Input is received in the form of N-best speech hypothesis lists. Each N-best hypothesis is passed through successive phases of parsing to logical form, ellipsis resolution, extraction of application-specific semantic representation, reference resolution, determination of dialogue response (including database search) and output generation. Both ellipsis resolution and reference resolution depend on the current dialogue state; all processing is completely side-effect free, and builds on ideas previously developed under NASA's PSA and Clarissa projects (Rayner et al 2000; Rayner et al 2005).

Processing of each N-best hypothesis results in a vector consisting of the forms produced at each of the different levels of representation, together with the confidence score assigned by the recognizer. These vectors are then rescored using a preference function which computes a weighted sum of feature scores, to select the final processing result. At present, we use four different features: the rank in the N-best hypothesis list, as computed by the recognizer; whether or not a dialogue move was produced by semantic processing; whether or not the tense of the main verb is consistent with other temporal expressions used; and whether or not the database query produced contains non-trivial constraints. Use of N-best rescoring reduces the semantic error rate by about 2% absolute, or 15% relative.

2.3 MRCP Server

The purpose of the MRCP server is to work as a mediator between the application and the ASR. It offers a layer of transparency and hides implementation details of proprietary integrations with different ASR and TTS

systems. This means that our architecture can potentially work with any system that supports the MRCP protocol, though we have so far only experimented with the Nuance MRCP server and the associated Nuance ASR and TTS systems.

2.4 Web Server

The final node of our system is the web server, which delivers recognition grammars or prerecorded audio files in response to requests from the MRCP server. The main benefit is that none of the target applications running on the device need to keep and update audio files and grammars; since they all share a common repository accessed through the web server, a change to the specific repository will be reflected to all the target applications.

3. The Calendar Application

The example Calendar application offers multi-modal access to a toy meeting database, and is available from the Open Source Regulus website. It is implemented on top of the Regulus platform, and can be run in either a desktop or a mobile device environment. Recognition currently uses a vocabulary of 211 surface words, of which about 50 are names extracted from the lexicon; language processing is context-dependent, with a mixed-initiative dialogue model. A short example interaction is shown in the table below.

User	Do you know what meetings there are next week?
System	[Recognition failed. Offer a list of help examples] Is there a meeting in the next week? What meetings are there during the next week? What meetings do we have next week? What meetings will there be next week? What meetings are there next week?
User	What meetings are there next week?
System	Meeting at Pierrette's room on September 3.
User	When is that meeting?
System	From 16:30 to 17:00 on September 3.
User	Will Marianne attend?
System	Do you mean Marianne Santaholma or Marianne Starlander?
User	Santaholma.
System	No.

Table 1: Interaction example with Calendar application

As seen in Figure 2 the Calendar GUI consists of three text boxes for presenting:

- The recognition result (grammar-based recognition).
- The result of the data query.
- A list of help examples associated with the recognition result (statistical recognition).

There is also a button for initiating the speech recognition and three buttons for navigating in the help 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.).

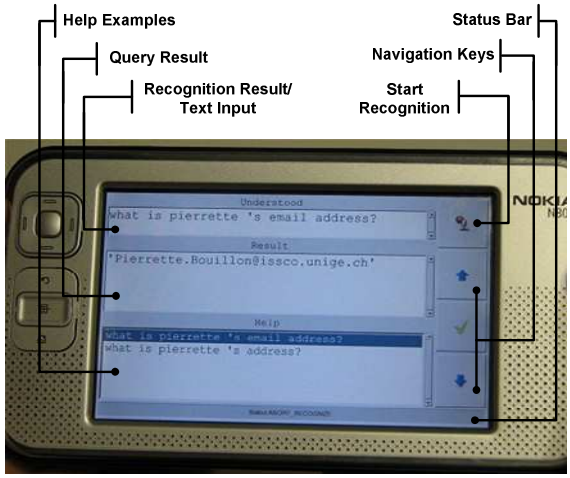

Figure 2: GUI components of the Calendar application

Following the normal Regulus application architecture, the system uses both a grammar-based language model (GLM) and a statistical language model (SLM); the former is used for main utterance processing, and the latter to drive the help system. Both the GLM and SLM models are constructed using tools from the Regulus platform. The base grammar for the GLM consists of the general feature grammar for English described in Chapter 9 of (Rayner et al 2006), together with an application-specific lexicon currently containing about 50 lemmas. In addition to this, there is a lexicon of names which is automatically created from the calendar database. The grammar is compiled using a set of semantics macros (Rayner et al, 2006, §7.5) which produce nested representations in which arguments are marked by their deep syntactic roles. This nested structure is necessary in order to handle constructions like "the next" or "the last"; for example, a simple attribute-value semantics would have great difficulty distinguishing "Did anyone from Geneva attend the last meeting at IDIAP?" from "Did anyone from IDIAP attend the last meeting at Geneva?".

A domain-specific feature grammar is extracted from the base grammar, using the corpus-driven specialization process described in Chapter 7 and 10 of (Rayner et al 2006). The initial training corpus consisted of about 200 utterances written by the developers. This enabled us to build a first running version of the system, after which all spoken input has been logged and transcribed. This data has been fed back into the training process, which has so far resulted in the addition of 400 more utterances to the training corpus. The recorded and transcribed data has also been used to drive development of the grammar and other rule sets. The current GLM contains 1650 CFG rules.

The SLM is a class trigram language model, built using the Nuance SayAnything © tool. The training corpus is the same as the one used for constructing the GLM. The backoff classes are specified using another Regulus tool, which allows each class to be defined with reference to syntactic properties of words in the Regulus lexicon.

4. System Evaluation

We evaluated the Calendar application to compare performance between the mobile platform and desktop versions. 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 eight speakers in an office environment. Each speakers had to read 50 selected, in coverage sentences during three interaction scenarios. We thus collected 150 sentences from each user, producing a total of 1200 waveforms (400 waveforms per interaction scenario). The scenarios were the following:

1. The user speaks to the desktop PC using a 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).

We consider that during a normal interaction the user would prefer to hold the device in front of him instead of putting it near to the ear like a telephone or moving it constantly near to his mouth. In this way he can interact with the system using not only his voice, but also seeing the output on the screen or picking a help example by the offered stylus pen.

In order to avoid favoring any particular scenario, the speakers read the sentences in different interaction orders. (We expected speakers to adapt to the device, and thus produce better results in the later sessions). The distribution of the speakers over the different interaction orders is shown in Table 2.

Configuration	Speakers	Waveform
MOB_H - DES_H - MOB_O	2	300
MOB_H - MOB_O - DES_H	2	300
DES_H - MOB_H - MOB_O	1	150
DES_H - MOB_O - MOB_H	1	150
MOB_O - DES_H - MOB_H	1	150
MOB_O - MOB_H - DES_H	1	150

Table 2: Speakers distribution by interaction order

The error rates for each interaction scenario are presented in Table 3. We present figures for three metrics: Word Error Rate (WER), Sentence Error Rate (SER) and Semantic Error Rate (SemER).

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 is thus in effect a version of SER that has been adjusted to take account of the fact that many recognition errors have no effect on system response.

	Desktop (DES_H)	Mobile (MOB_H)	Mobile (MOB_O)
WER	13.05%	12.83%	21.21%
SemER	22.8%	21.9%	33.9%
SER	42.25%	44.08%	55.7%

Table 3: Error Rates per interaction scenario

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. 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.

We should in passing say a few words about the specific values for the error rates. As usual, both SER and SemER are substantially greater than WER. This is to be expected, given that a single mistake in a sentence can change its semantic meaning. For example, if we recognize: "Will there be a meeting on the *fifth* of July?" instead of: "Will there be a meeting on the *fifteenth* of July?" both SER and SemER will count the whole example as incorrect, but WER will only count one substitution error in ten words.

The high absolute values for WER and SemER are quite surprising as in other domains with similar vocabulary sizes, Regulus applications have typically delivered WER around 4-8% and SemER around 5-10% (Bouillon et al 2007, Chapter 11 of Rayner et al 2006, Rayner et al 2005). Hand-examination of detailed results from the tests suggested that the poor results are due to the domain being unexpectedly challenging, despite its modest vocabulary. There are several common pairs of words which are easily confusable. For example, "when" and "where" sound similar and have almost identical distributions, but result in different semantic forms. ("When is the meeting?" versus "Where is the meeting?") Still worse is the fact that the articles "a" and "the" frequently have semantic content, which is unusual for a medium-vocabulary task.

For example, “Was Pierrette at *the* meeting in Geneva?” asks about Pierrette's attendance at a specific meeting in Geneva, which needs to be determined from preceding context by reference resolution; however “Was Pierrette at *a* meeting in Geneva?” asks about her attendance at any meeting held in Geneva. Similarly, “Give me meetings for next week” asks for meetings in the seven day period starting next Monday, while “Give me meetings for *the* next week” asks for meetings in the seven day period starting today.

5. Summary and conclusion

We have described how we were able to extend the Open Source Regulus platform to permit hosting of speech-enabled Regulus applications on mobile devices. The infrastructure is based on a distributed architecture which uses state-of-the-art integration techniques to combine pre-existing Regulus resources with commercial ASR systems.

We performed an initial proof-of-concept evaluation of the architecture, using a calendar application with a vocabulary of about 200 words. The application's performance on the mobile platform was essentially identical to that on a standard desktop PC.

In future work, we plan to use the implemented infrastructure to create more challenging applications. Evaluation will not be constrained to the office environment, but will be extended to outdoor testing under different conditions. These will at a minimum include variation in ambient noise level and network traffic load.

6. Acknowledgements

Part of this work was developed in the context of the Interactive Multimodal Information Management project (IM2.HMI) funded by the Swiss National Science Foundation.

7. References

Bouillon P., Rayner M., Novellas Vall, B., Starlander M., Santaholma M., Nakao Y. and N. Chatzichrisafis (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.

Bouillon, P., F. Ehsani, R. Frederking and M. Rayner (Eds.) (2006). Medical SpeechTranslation - Proceedings of the Workshop. HLT/NAACL-06, New York, NY, USA.

Bouillon, P., M. Rayner, N. Chatzichrisafis, B. A. Hockey, M. Santaholma, M. Starlander, H. Isahara, K. Kanzaki and Y. Nakao (2005). A Generic Multi-Lingual Open Source Platform for Limited-Domain Medical Speech Translation. Proceedings of the 10th Conference of the European Association of Machine Translation, Budapest, Hungary.

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. Proceedings of the First International Workshop on Medical Speech Translation, HLT-NAACL, New York.

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

Rayner, M., B. A. Hockey, N. Chatzichrisafis, K. Farrell and J.-M. Renders (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.

Rayner, M., D. Carter, P. Bouillon, V. Digalakis and M. Wirén (2000). The Spoken Language Translator. Cambridge, Cambridge University Press.

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

Starlander, M., P. Bouillon, N. Chatzichrisafis, M. Santaholma, M. Rayner, B. A. Hockey, H. Isahara, K. Kanzaki and Y. Nakao (2005). Practicing Controlled Language through a Help System integrated into the Medical Speech Translation System (MedSLT). Proceedings of the 10th Machine Translation Summit (MT Summit X), Phuket, Thailand.

Tsourakis, N., D. Pratsolis, C. Harizakis and V. Digalakis (2006). An Architecture for Multimodal Applications over Wireless Data Networks. Proceedings of the IET International Conference on Intelligent Environents, Athens, Greece.

Waibel, A., A. Badran, A. Black, R. Frederking, D. Gates, A. Lavie, L. Levin, K. Lenzo, L. Tomokiyo, J. Reichert, T. Schultz, D. Wallace, M. Woszczyna and J. Zhang (2003). Speechalator: Two-Way Speech-To-Speech Translation In Your Hand. Proceedings of the 8th European Conference on Speech Communication and Technology, Geneva, Switzerland.