ACL-08: HLT

Workshop on Mobile Language Processing

Proceedings of the Workshop

June 20, 2008
The Ohio State University
Columbus, Ohio, USA

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53707
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-13-8

Introduction

Mobile devices such as ultra-mobile PCs, personal digital assistants, and smart phones have many unique characteristics that make them both highly desirable as well as difficult to use. On the positive side, they are small, convenient, personalizable, and provide an anytime-anywhere communication capability. On the other hand, they have limited input and output capabilities, limited bandwidth, limited memory, and restricted processing power.

In anticipation of new and exciting applications for natural and spoken language processing on mobile devices, this workshop provided a forum for discussing some of the challenges that are unique to this domain. For instance, mobile devices are beginning to integrate sensors (most commonly for location detection through GPS, Global Positioning Systems) that can be exploited by context/location aware NLP systems. Another interesting research direction is the use of information from multiple devices for "distributed" language modeling and inference. To give some concrete examples, knowing the type of web queries made from nearby devices or from a specific location or 'context' can be combined for various applications and could potentially improve information retrieval results. Learned language models could be transferred from device to device, propagating and updating the language models continuously and in a decentralized manner.

Processing and memory limitations faced by the execution of NLP and speech recognition software on small devices need to be addressed. Several papers addressed this issue. In *"Information extraction using finite state automata and syllable n-grams"* Seon et al. proposed a modified HMM for information extraction in a mobile environment. This kind of model has the advantage of being compact. Huggins-Daines et al. proposed a simple entropy-based technique to improve the scalability of acoustic models in embedded systems; they showed a significant speed-up in recognition with a negligible increase in word error rate (*"Mixture Pruning and Roughening for Scalable Acoustic Models."*) Ganchev and Dredze in *"Small Statistical Models by Random Feature Mixing"* showed how it is possible to do efficient NLP learning by reducing the number of parameters on resource constrained devices with little loss in performance; and *"A Wearable Headset Speech-to-Speech Translation System"* by Krstovski et al. shrunk a speech translation system to fit into a wearable speech-to-speech translation system.

Some applications and practical considerations may require a client/server or distributed architecture: what are the implications for language processing systems in using such architectures? Homola (*"A Distributed Database for Mobile NLP Applications"*) proposed a distributed database for lexical transfer in machine translation. The database contains data shared among multiple devices and automatically synchronizes them.

The limitation of the input and output channels necessitates typing on increasingly smaller keyboards which can be quite difficult, and similarly reading on small displays is challenging. Speech interfaces for dictation or for understanding navigation commands and/or language models for typing suggestions would enhance the input channel, while NLP systems for text classification, summarization and

information extraction would be helpful for the output channel. Speech and multimodal interfaces, language generation and dialog systems would provide a natural way to interact with mobile devices. A multimodal dialogue system for interacting with a home entertainment center via a mobile device was proposed by Gruenstein et al. in *"A Multimodal Home entertainment Interface via a Mobile Device."*

Furthermore, the growing market of cell phones in developing regions can be used for delivering applications in the areas of health, education and economic growth to rural communities. Some of the challenges in this area are the limited literacy, the many languages and dialects spoken and the networking infrastructure.

For the health domain, Nikolova and Ma in their paper *"Assistive Mobile Communication Support"* discussed the role of mobile technologies in a system for communication support for people with speech and language disabilities.

We believe that the issues raised by the papers in this Workshop represent just the tip of the iceberg, and we hope that by raising awareness of these issues, more research will be aimed at mobile language processing. The ACL 2008 Workshop on Mobile Language Processing took place on June 20 in Columbus, Ohio following ACL-08: HLT with an invited talk by Dr. Lisa Stifelman, Principal User Experience Manager at Tellme/Microsoft, seven oral paper presentations, a poster and a demo session and a panel discussion.

We thank the members of the Program Committee for their diligent and insightful reviews, as well as our illustrious Panel Session members.

Barbara Rosario and Tim Paek
Co-Organizers

Organizers:

Barbara Rosario, Intel Research
Tim Paek, Microsoft Research

Program Committee:

Alex Acero, Microsoft Research
Alan Black, CMU
Dilek Hakkani Tur, ICSI
Marti Hearst, iSchool, UC Berkeley
Michael Johnston, AT&T
Maryam Kamvar, Google and Columbia University
Kevin Knight, USC/Information Sciences Institute
Julian Kupiec, Google
Dekang Lin, University of Alberta, Canada
Maryam Mahdaviani, University of British Columbia, Canada
Wolfgang Minker, University of Ulm, Germany
Noah Smith, CMU
Bo Thiesson, Microsoft Research
Gokhan Tur , SRI
Fuliang Weng, Bosch
Thomas Zheng , Tsinghua University
Geoffrey Zweig, Microsoft Research

Invited Speaker:

Lisa Stifelman, Principal User Experience Manager at Tellme/Microsoft.

Table of Contents

Workshop Program

Friday, June 20, 2008

8:45–9:00 Opening Remarks

9:00–10:00 Invited Talk by Dr. Lisa Stifelman, Principal User Experience Manager at Tellme/Microsoft. *Say it and See it! Applying User-Centered Design to Mobile and Multimodal Search.*

10:00–10:30 *A Multimodal Home Entertainment Interface via a Mobile Device*
Alexander Gruenstein, Bo-June (Paul) Hsu, James Glass, Stephanie Seneff, Lee Hetherington, Scott Cyphers, Ibrahim Badr, Chao Wang and Sean Liu

10:30–11:00 Break

11:00–11:25 *A Wearable Headset Speech-to-Speech Translation System*
Kriste Krstovski, Michael Decerbo, Rohit Prasad, David Stallard, Shirin Saleem and Premkumar Natarajan

11:25–11:50 *Information extraction using finite state automata and syllable n-grams in a mobile environment*
Choong-Nyoung Seon, Harksoo Kim and Jungyun Seo

11:50–12:15 *Small Statistical Models by Random Feature Mixing*
Kuzman Ganchev and Mark Dredze

12:15–1:15 Lunch

1:15–1:40 *Mixture Pruning and Roughening for Scalable Acoustic Models*
David Huggins-Daines and Alexander I. Rudnicky

1:40–2:05 *Assistive Mobile Communication Support*
Sonya Nikolova and Xiaojuan Ma

2:05–2:30 *A Distributed Database for Mobile NLP Applications*
Petr Homola

2:30–3:30 Demos and Posters

3:30–4:00 Break

4:00–5:00 Panel Session

A Multimodal Home Entertainment Interface via a Mobile Device

Alexander Gruenstein Bo-June (Paul) Hsu James Glass Stephanie Seneff
Lee Hetherington Scott Cyphers Ibrahim Badr Chao Wang Sean Liu
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar St, Cambridge, MA 02139 USA
http://www.sls.csail.mit.edu/

Abstract

We describe a multimodal dialogue system for interacting with a home entertainment center via a mobile device. In our working prototype, users may utilize both a graphical and speech user interface to search TV listings, record and play television programs, and listen to music. The developed framework is quite generic, potentially supporting a wide variety of applications, as we demonstrate by integrating a weather forecast application. In the prototype, the mobile device serves as the locus of interaction, providing both a small touch-screen display, and speech input and output; while the TV screen features a larger, richer GUI. The system architecture is agnostic to the location of the natural language processing components: a consistent user experience is maintained regardless of whether they run on a remote server or on the device itself.

1 Introduction

People have access to large libraries of digital content both in their living rooms and on their mobile devices. Digital video recorders (DVRs) allow people to record TV programs from hundreds of channels for subsequent viewing at home—or, increasingly, on their mobile devices. Similarly, having accumulated vast libraries of digital music, people yearn for an easy way to sift through them from the comfort of their couches, in their cars, and on the go.

Mobile devices are already central to *accessing* digital media libraries while users are away from home: people listen to music or watch video recordings. Mobile devices also play an increasingly important role in *managing* digital media libraries. For instance, a web-enabled mobile phone can be used to remotely schedule TV recordings through a web site or via a custom application. Such management tasks often prove cumbersome, however, as it is challenging to browse through listings for hundreds of TV channels on a small display. Indeed, even on a large screen in the living room, browsing alphabetically, or by time and channel, for a particular show using the remote control quickly becomes unwieldy.

Speech and multimodal interfaces provide a natural means of addressing many of these challenges. It is effortless for people to say the name of a program, for instance, in order to search for existing recordings. Moreover, such a speech browsing capability is useful both in the living room and away from home. Thus, a natural way to provide speech-based control of a media library is through the user's mobile device itself.

In this paper we describe just such a prototype system. A mobile phone plays a central role in providing a multimodal, natural language interface to both a digital video recorder and a music library. Users can interact with the system—presented as a dynamic web page on the mobile browser—using the navigation keys, the stylus, or spoken natural language. In front of the TV, a much richer GUI is also available, along with support for playing video recordings and music.

In the prototype described herein, the mobile device serves as the locus of natural language interaction, whether a user is in the living room or walking down the street. Since these environments may be very different in terms of computational re-

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 1–8,
Columbus, Ohio, USA, June 2008.

sources and network bandwidth, it is important that the architecture allows for multiple configurations in terms of the *location* of the natural language processing components. For instance, when a device is connected to a Wi-Fi network at home, recognition latency may be reduced by performing speech and natural language processing on the home media server. Moreover, a powerful server may enable more sophisticated processing techniques, such as multipass speech recognition (Hetherington, 2005; Chung et al., 2004), for improved accuracy. In situations with reduced network connectivity, latency may be improved by performing speech recognition and natural language processing tasks on the mobile device itself. Given resource constraints, however, less detailed acoustic and language models may be required. We have developed just such a flexible architecture, with many of the natural language processing components able to run on either a server or the mobile device itself. Regardless of the configuration, a consistent user experience is maintained.

2 Related Work

Various academic researchers and commercial businesses have demonstrated speech-enabled interfaces to entertainment centers. A good deal of the work focuses on adding a microphone to a remote control, so that speech input may be used in addition to a traditional remote control. Much commercial work, for example (Fujita et al., 2003), tends to focus on constrained grammar systems, where speech input is limited to a small set of templates corresponding to menu choices. (Berglund and Johansson, 2004) present a remote-control based speech interface for navigating an existing interactive television on-screen menu, though experimenters manually transcribed user utterances as they spoke instead of using a speech recognizer. (Oh et al., 2007) present a dialogue system for TV control that makes use of concept spotting and statistical dialogue management to understand queries. A version of their system can run independently on low-resource devices such as PDAs; however, it has a smaller vocabulary and supports a limited set of user utterance templates. Finally, (Wittenburg et al., 2006) look mainly at the problem of searching for television programs using speech, an on-screen display, and a remote control. They explore a Speech-In List-Out interface to searching for episodes of television programs.

(Portele et al., 2003) depart from the model of adding a speech interface component to an existing on-screen menu. Instead, they a create a tablet PC interface to an electronic program guide, though they do not use the television display as well. Users may search an electronic program guide using constraints such as date, time, and genre; however, they can't search by title. Users can also perform typical remote-control tasks like turning the television on and off, and changing the channel. (Johnston et al., 2007) also use a tablet PC to provide an interface to television content—in this case a database of movies. The search can be constrained by attributes such as title, director, or starring actors. The tablet PC pen can be used to handwrite queries and to point at items (such as actor names) while the user speaks.

We were also inspired by previous prototypes in which mobile devices have been used in conjunction with larger, shared displays. For instance, (Paek et al., 2004) demonstrate a framework for building such applications. The prototype we demonstrate here fits into their "Jukebox" model of interaction. Interactive workspaces, such as the one described in (Johanson et al., 2002), also demonstrate the utility of integrating mobile and large screen displays. Our prototype is a departure from these systems, however, in that it provides for spoken interactions.

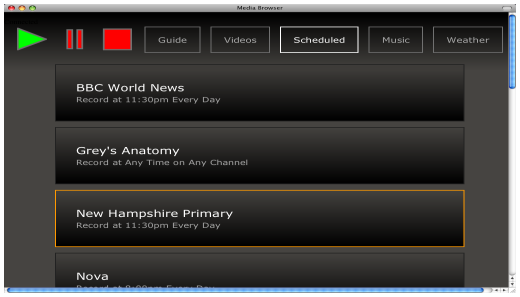
Finally, there is related work in the use of mobile devices for various kinds of search. For instance, offerings from Microsoft (Acero et al., 2008), Vlingo,[1] and Promptu[2] allow users to search for items like businesses and songs using their mobile phones. These applications differ from ours in that speech is used only for search, without any accompanying command and control capabilities. Also, these services do not allow interaction with your own devices at home. Efforts have been made to use mobile devices for control of devices in the home, such as in (Nichols and Myers, 2006), however these efforts have not involved the use of speech as an input modality.

[1] http://www.vlingo.com

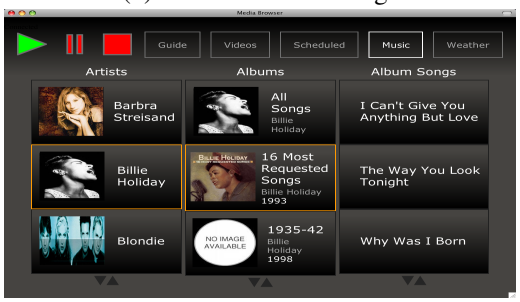
[2] http://www.promptu.com

Figure 1: User interface overview.

(a) Program Guide



(b) Scheduled Recordings



(c) Music Library

U: What's on tonight at seven o'clock?
S: I found seventy entries. [*Displays program guide.*]
U: Are there any *action movies* on *HBO* tomorrow?
S: I found two entries. [*Displays results.*]
U: When is *American Idol* on?
S: I found three entries. [*Displays results.*]
U: Record the first one.
S: I have added *American Idol* to the list of programs to be recorded. [*Displays recording settings.*]
U: Please record *Nova* on Wednesday night.
S: I have added *Nova* to the list of programs to be recorded. [*Displays recording settings.*]
U: Show all my scheduled recordings.
S: There are seven scheduled recordings. [*Shows list.*]
U: Delete *BBC World News*.
S: I have deleted the scheduled recording: *BBC World News*. [*Updates List.*]
U: Show my videos.
S: [*Displays list of videos.*]
U: Delete *Frasier*.
S: I have deleted the recording *Frasier*. [*Updates list.*]
U: Play *Grey's Anatomy*.
S: Playing *Grey's Anatomy*. [*Plays recording.*]
U: Stop Playing.
S: Stopping. [*Stops playback.*]
U: [*Clicks on Music tab.*]
S: [*Displays artist list.*]
U: Show albums by *Billie Holiday*.
S: I found three albums by *Billie Holiday*. [*Shows albums.*]
U: Please play *A Hard Rain's A-Gonna Fall* by *Bob Dylan*.
S: Playing *A Hard Rain's A-Gonna Fall*. [*Plays song.*]

Figure 2: Screenshots and an example interaction. Utterances are labeled with *U* for user and *S* for system.

3 User Experience

Our current prototype system implements the basic functionalities that one expects from a home entertainment center. Users can navigate through and record programs from the television's electronic program guide, manage recording settings, and play recorded videos. They can also browse and listen to selections from their music libraries. However, unlike existing prototypes, ours employs a smartphone with a navigation pad, touch-sensitive screen, and built-in microphone as the remote control. Figure 1 provides an overview of the graphical user interface on both the TV and mobile device.

Mirroring the TV's on-screen display, the prototype system presents a reduced view on the mobile device with synchronized cursors. Users can navigate the hierarchical menu structure using the arrow keys or directly click on the target item with the stylus. While away from the living room, or when a recording is playing full screen, users can browse and manage their media libraries using only the mobile device.

While the navigation pad and stylus are great for basic navigation and control, searching for media with specific attributes, such as title, remains cumbersome. To facilitate such interactions, the current system supports spoken natural language interactions. For example, the user can press the hold-to-talk button located on the side of the mobile device and ask "What's on the *National Geographic Channel* this afternoon?" to retrieve a list of shows with the specified channel and time. The system responds with a short verbal summary "I found six entries on January seventh" and presents the resulting list on both the TV and mobile displays. The user can then browse the list using the navigation pad or press the hold-to-talk button to barge in with another command, *e.g.* "Please record the second one." Depressing the hold-to-talk button not only terminates any current spoken response, but also mutes the TV to minimize interference with speech recognition. As the previous example demonstrates, contextual information is used to resolve list position references and disambiguate commands.

The speech interface to the user's music library works in a similar fashion. Users can search by artist, album, and song name, and then play the songs found. To demonstrate the extensibility of the architecture, we have also integrated an existing weather information system (Zue et al., 2000), which has been previously deployed as a telephony application. Users simply click on the *Weather* tab to switch to this domain, allowing them to ask a wide range of weather queries. The system responds verbally and with a simple graphical forecast.

To create a natural user experience, we designed the multimodal interface to allow users to seamlessly switch among the different input modalities available on the mobile device. Figure 2 demonstrates an example interaction with the prototype, as well as several screenshots of the user interface.

4 System Architecture

The system architecture is quite flexible with regards to the placement of the natural language processing components. Figure 3 presents two possible configurations of the system components distributed across the mobile device, home media server, and TV display. In 3(a), all speech recognition and natural language processing components reside on the server, with the mobile device acting as the microphone, speaker, display, and remote control. In 3(b), the speech recognizer, language understanding component, language generation component, and text-to-speech (TTS) synthesizer run on the mobile device. Depending on the capabilities of the mobile device and network connection, different configurations may be optimal. For instance, on a powerful device with slow network connection, recognition latency may be reduced by performing speech recognition and natural language processing on the device. On the other hand, streaming audio via a fast wireless network to the server for processing may result in improved accuracy.

In the prototype system, flexible and reusable speech recognition and natural language processing capabilities are provided via generic components developed and deployed in numerous spoken dialogue systems by our group, with the exception of an off-the-shelf speech synthesizer. Speech input from the mobile device is recognized using the landmark-based SUMMIT system (Glass, 2003). The resulting N-best hypotheses are processed by the TINA language understanding component (Seneff, 1992).

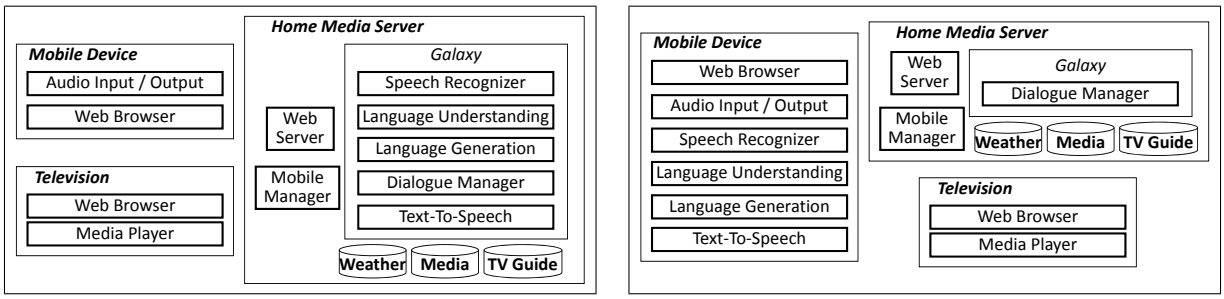


Figure 3: Two architecture diagrams. In (a) speech recognition and natural language processing occur on the server, while in (b) processing is primarily performed on the device.

Based on the resulting meaning representation, the dialogue manager (Polifroni et al., 2003) incorporates contextual information (Filisko and Seneff, 2003), and then determines an appropriate response. The response consists of an update to the graphical display, and a spoken system response which is realized via the GENESIS (Baptist and Seneff, 2000) language generation module. To support on-device processing, all the components are linked via the GALAXY framework (Seneff et al., 1998) with an additional *Mobile Manager* component responsible for coordinating the communication between the mobile device and the home media server.

In the currently deployed system, we use a mobile phone with a 624 MHz ARM processor running the Windows Mobile operating system and Opera Mobile web browser. The TV program and music databases reside on the home media server running GNU/Linux. The TV program guide data and recording capabilities are provided via MythTV, a full-featured, open-source digital video recorder software package.[3] Daily updates to the program guide information typically contain hundreds of unique channel names and thousands of unique program names. The music library is comprised of 5,000 songs from over 80 artists and 13 major genres, indexed using the open-source text search engine Lucene.[4] Lastly, the TV display can be driven by a web browser on either the home media server or a separate computer connected to the server via a fast Ethernet connection, for high quality video streaming.

[3] `http://www.mythtv.org/`

[4] `http://lucene.apache.org/`

While the focus of this paper is on the natural language processing and user interface aspects of the system, our work is actually situated within a larger collaborative project at MIT that also includes simplified device configuration (Mazzola Paluska et al., 2008; Mazzola Paluska et al., 2006), transparent access to remote servers (Ford et al., 2006), and improved security.

5 Mobile Natural Language Components

Porting the implementation of the various speech recognizer and natural language processing components to mobile devices with limited computation and memory presents both a research and engineering challenge. Instead of creating a small vocabulary, fixed phrase dialogue system, we aim to support—on the mobile device—the same flexible and natural language interactions currently available on our desktop, tablet, and telephony systems; see *e.g.*, (Gruenstein et al., 2006; Seneff, 2002; Zue et al., 2000). In this section, we summarize our efforts thus far in implementing the SUMMIT speech recognizer and TINA natural language parser. Ports of the GENESIS language generation system and of our dialogue manager are well underway, and we expect to have these components working on the mobile device in the near future.

5.1 PocketSUMMIT

To significantly reduce the memory footprint and overall computation, we chose to reimplement our segment-based speech recognizer from scratch, utilizing fixed-point arithmetic, parameter quantization, and bit-packing in the binary model files. The resulting PocketSUMMIT recognizer (Hether-

ington, 2007) utilizes only the landmark features, initially forgoing segment features such as phonetic duration, as they introduce algorithmic complexities for relatively small word error rate (WER) improvements.

In the current system, we quantize the mean and variance of each Gaussian mixture model dimension to 5 and 3 bits, respectively. Such quantization not only results in an 8-fold reduction in model size, but also yields about a 50% speedup by enabling table lookups for Gaussian evaluations. Likewise, in the finite-state transducers (FSTs) used to represent the language model, lexical, phonological, and class diphone constraints, quantizing the FST weights and bit-packing not only compress the resulting binary model files, but also reduce the processing time with improved processor cache locality.

In the aforementioned TV, music, and weather domains with a moderate vocabulary of a few thousand words, the resulting PocketSUMMIT recognizer performs in approximately real-time on 400-600 MHz ARM processors, using a total of 2-4 MB of memory, including 1-2 MB for memory-mapped model files. Compared with equivalent non-quantized models, PocketSUMMIT achieves dramatic improvements in speed and memory while maintaining comparable WER performance.

5.2 PocketTINA

Porting the TINA natural language parser to mobile devices involved significant software engineering to reduce the memory and computational requirements of the core data structures and algorithms. TINA utilizes a best-first search that explores thousands of partial parses when processing an input utterance. To efficiently manage memory allocation given the unpredictability of pruning invalid parses (*e.g.* due to subject-verb agreement), we implemented a mark and sweep garbage collection mechanism. Combined with a more efficient implementation of the priority queue and the use of aggressive "beam" pruning, the resulting PocketTINA system provides identical output as server-side TINA, but can parse a 10-best recognition hypothesis list into the corresponding meaning representation in under 0.1 seconds, using about 2 MB of memory.

6 Rapid Dialogue System Development

Over the course of developing dialogue systems for many domains, we have built generic natural language understanding components that enable the rapid development of flexible and natural spoken dialogue systems for novel domains. Creating such prototype systems typically involves customizing the following to the target domain: recognizer language model, language understanding parser grammar, context resolution rules, dialogue management control script, and language generation rules.

Recognizer Language Model Given a new domain, we first identify a set of semantic classes which correspond to the back-end application's database, such as *artist*, *album*, and *genre*. Ideally, we would have a corpus of tagged utterances collected from real users. However, when building prototypes such as the one described here, little or no training data is usually available. Thus, we create a domain-specific context-free grammar to generate a supplemental corpus of synthetic utterances. The corpus is used to train probabilities for the natural language parsing grammar (described immediately below), which in turn is used to derive a class n-gram language model (Seneff et al., 2003).

Classes in the language model which correspond to contents of the database are marked as dynamic, and are populated at runtime from the database (Chung et al., 2004; Hetherington, 2005). Database entries are heuristically normalized into spoken forms. Pronunciations not in our 150,000 word lexicon are automatically generated (Seneff, 2007).

Parser Grammar The TINA parser uses a probabilistic context-free grammar enhanced with support for wh-movement and grammatical agreement constraints. We have developed a generic syntactic grammar by examining hundreds of thousands of utterances collected from real user interactions with various existing dialogue systems. In addition, we have developed libraries which parse and interpret common semantic classes like dates, times, and numbers. The grammar and semantic libraries provide good coverage for spoken dialogue systems in database-query domains.

To build a grammar for a new domain, a developer extends the generic syntactic grammar by augmenting it with domain-specific semantic categories and their lexical entries. A probability model which conditions each node category on its left sibling and parent is then estimated from a training corpus of utterances (Seneff et al., 2003).

At runtime, the recognizer tags the hypothesized dynamic class expansions with their class names, allowing the parser grammar to be independent of the database contents. Furthermore, each semantic class is designated either as a semantic *entity*, or as an *attribute* associated with a particular *entity*. This enables the generation of a semantic representation from the parse tree.

Dialogue Management & Language Generation Once an utterance is recognized and parsed, the meaning representation is passed to the context resolution and dialogue manager component. The context resolution module (Filisko and Seneff, 2003) applies generic and domain-specific rules to resolve anaphora and deixis, and to interpret fragments and ellipsis in context. The dialogue manager then interacts with the application back-end and database, controlled by a script customized for the domain (Polifroni et al., 2003). Finally, the GENESIS module (Baptist and Seneff, 2000) applies domain-specific rules to generate a natural language representation of the dialogue manager's response, which is sent to a speech synthesizer. The dialogue manager also sends an update to the GUI, so that, for example, the appropriate database search results are displayed.

7 Mobile Design Challenges

Dialogue systems for mobile devices present a unique set of design challenges not found in telephony and desktop applications. Here we describe some of the design choices made while developing this prototype, and discuss their tradeoffs.

7.1 Client/Server Tradeoffs

Towards supporting network-less scenarios, we have begun porting various natural language processing components to mobile platforms, as discussed in Section 5. Having efficient mobile implementations further allows the natural language processing tasks to be performed on either the mobile device or the server. While building the prototype, we observed that the Wi-Fi network performance can often be unpredictable, resulting in erratic recognition latency that occasionally exceeds on-device recognition latency. However, utilizing the mobile processor for computationally intensive tasks rapidly drains the battery. Currently, the component architecture in the prototype system is pre-configured. A more robust implementation would dynamically adjust the configuration to optimize the tradeoffs among network use, CPU utilization, power consumption, and user-perceived latency/accuracy.

7.2 Speech User Interface

As neither open-mic nor push-to-talk with automatic endpoint detection is practical on mobile devices with limited battery life, our prototype system employs a hold-to-talk hardware button for microphone control. To guide users to speak commands only while the button is depressed, a short beep is played as an earcon both when the button is pushed and released. Since users are less likely to talk over short audio clips, the use of earcons mitigates the tendency for users to start speaking before pushing down the microphone button.

In the current system, media audio is played over the TV speakers, whereas TTS output is sent to the mobile device speakers. To reduce background noise captured from the mobile device's far-field microphone, the TV is muted while the microphone button is depressed. Unlike telephony spoken dialogue systems where the recognizer has to constantly monitor for barge-in, the use of a hold-to-talk button significantly simplifies barge-in support, while reducing power consumption.

7.3 Graphical User Interface

In addition to supporting interactive natural language dialogues via the spoken user interface, the prototype system implements a graphical user interface (GUI) on the mobile device to supplement the TV's on-screen interface. To faciliate rapid prototyping, we chose to implement both the mobile and TV GUI using web pages with AJAX (Asynchronous Javascript and XML) techniques, an approach we have leveraged in several existing multimodal dialogue systems, *e.g.* (Gruenstein et al.,

2006; McGraw and Seneff, 2007). The resulting interface is largely platform-independent and allows display updates to be "pushed" to the client browser.

As many users are already familiar with the TV's on-screen interface, we chose to mirror the same interface on the mobile device and synchronize the selection cursor. However, unlike desktop GUIs, mobile devices are constrained by a small display, limited computational power, and reduced network bandwidth. Thus, both the page layout and information detail were adjusted for the mobile browser. Although AJAX is more responsive than traditional web technology, rendering large formatted pages—such as the program guide grid—is often still unacceptably slow. In the current implementation, we addressed this problem by displaying only the first section of the content and providing a "Show More" button that downloads and renders the full content. While browser-based GUIs expedite rapid prototyping, deployed systems may want to take advantage of native interfaces specific to the device for more responsive user interactions. Instead of limiting the mobile interface to reflect the TV GUI, improved usability may be obtained by designing the interface for the mobile device first and then expanding the visual content to the TV display.

7.4 Client/Server Communication

In the current prototype, communication between the mobile device and the media server consists of AJAX HTTP and XML-RPC requests. To enable server-side "push" updates, the client periodically pings the server for messages. While such an implementation provides a responsive user interface, it quickly drains the battery and is not robust to network outages resulting from the device being moved or switching to power-saving mode. Reestablishing connection with the server further introduces latency. In future implementations, we would like to examine the use of Bluetooth for lower power consumption, and infrared for immediate response to common controls and basic navigation.

8 Conclusions & Future Work

We have presented a prototype system that demonstrates the feasibility of deploying a multimodal, natural language interface on a mobile device for browsing and managing one's home media library. In developing the prototype, we have experimented with a novel role for a mobile device—that of a speech-enabled remote control. We have demonstrated a flexible natural language understanding architecture, in which various processing stages may be performed on either the server or mobile device, as networking and processing power considerations require.

While the mobile platform presents many challenges, it also provides unique opportunities. Whereas desktop computers and TV remote controls tend to be shared by multiple users, a mobile device is typically used by a single individual. By collecting and adapting to the usage data, the system can personalize the recognition and understanding models to improve the system accuracy. In future systems, we hope to not only explore such adaptation possibilities, but also study how real users interact with the system to further improve the user interface.

Acknowledgments

This research is sponsored by the TParty Project, a joint research program between MIT and Quanta Computer, Inc.; and by Nokia, as part of a joint MIT-Nokia collaboration. We are also thankful to three anonymous reviewers for their constructive feedback.

References

A. Acero, N. Bernstein, R. Chambers, Y. C. Jui, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweig. 2008. Live search for mobile: Web services by voice on the cellphone. In *Proc. of ICASSP*.

L. Baptist and S. Seneff. 2000. Genesis-II: A versatile system for language generation in conversational system applications. In *Proc. of ICSLP*.

A. Berglund and P. Johansson. 2004. Using speech and dialogue for interactive TV navigation. *Universal Access in the Information Society*, 3(3-4):224–238.

G. Chung, S. Seneff, C. Wang, and L. Hetherington. 2004. A dynamic vocabulary spoken dialogue interface. In *Proc. of INTERSPEECH*, pages 327–330.

E. Filisko and S. Seneff. 2003. A context resolution server for the GALAXY conversational systems. In *Proc. of EUROSPEECH*.

B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. 2006. Persistent personal names for globally connected mobile devices. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*.

K. Fujita, H. Kuwano, T. Tsuzuki, and Y. Ono. 2003. A new digital TV interface employing speech recognition. *IEEE Transactions on Consumer Electronics*, 49(3):765–769.

J. Glass. 2003. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17:137–152.

A. Gruenstein, S. Seneff, and C. Wang. 2006. Scalable and portable web-based multimodal dialogue interaction with geographical databases. In *Proc. of INTERSPEECH*.

I. L. Hetherington. 2005. A multi-pass, dynamic-vocabulary approach to real-time, large-vocabulary speech recognition. In *Proc. of INTERSPEECH*.

I. L. Hetherington. 2007. PocketSUMMIT: Small-footprint continuous speech recognition. In *Proc. of INTERSPEECH*, pages 1465–1468.

B. Johanson, A. Fox, and T. Winograd. 2002. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2):67–74.

M. Johnston, L. F. D'Haro, M. Levine, and B. Renger. 2007. A multimodal interface for access to content in the home. In *Proc. of ACL*, pages 376–383.

J. Mazzola Paluska, H. Pham, U. Saif, C. Terman, and S. Ward. 2006. Reducing configuration overhead with goal-oriented programming. In *PerCom Workshops*, pages 596–599. IEEE Computer Society.

J. Mazzola Paluska, H. Pham, U. Saif, G. Chau, C. Terman, and S. Ward. 2008. Structured decomposition of adapative applications. In *Proc. of 6th IEEE Conference on Pervasive Computing and Communications*.

I. McGraw and S. Seneff. 2007. Immersive second language acquisition in narrow domains: A prototype ISLAND dialogue system. In *Proc. of the Speech and Language Technology in Education Workshop*.

J. Nichols and B. A. Myers. 2006. Controlling home and office appliances with smartphones. *IEEE Pervasive Computing, special issue on SmartPhones*, 5(3):60–67, July-Sept.

H.-J. Oh, C.-H. Lee, M.-G. Jang, and Y. K. Lee. 2007. An intelligent TV interface based on statistical dialogue management. *IEEE Transactions on Consumer Electronics*, 53(4).

T. Paek, M. Agrawala, S. Basu, S. Drucker, T. Kristjansson, R. Logan, K. Toyama, and A. Wilson. 2004. Toward universal mobile interaction for shared displays. In *Proc. of Computer Supported Cooperative Work*.

J. Polifroni, G. Chung, and S. Seneff. 2003. Towards the automatic generation of mixed-initiative dialogue systems from web content. In *Proc. EUROSPEECH*, pages 193–196.

T. Portele, S. Goronzy, M. Emele, A. Kellner, S. Torge, and J. te Vrugt. 2003. SmartKom-Home - an advanced multi-modal interface to home entertainment. In *Proc. of INTERSPEECH*.

S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue. 1998. GALAXY-II: A reference architecture for conversational system development. In *Proc. ICSLP*.

S. Seneff, C. Wang, and T. J. Hazen. 2003. Automatic induction of n-gram language models from a natural language grammar. In *Proceedings of EUROSPEECH*.

S. Seneff. 1992. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86.

S. Seneff. 2002. Response planning and generation in the MERCURY flight reservation system. *Computer Speech and Language*, 16:283–312.

S. Seneff. 2007. Reversible sound-to-letter/letter-to-sound modeling based on syllable structure. In *Proc. of HLT-NAACL*.

K. Wittenburg, T. Lanning, D. Schwenke, H. Shubin, and A. Vetro. 2006. The prospects for unrestricted speech input for TV content search. In *Proc. of AVI'06*.

V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. 2000. JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1), January.

A Wearable Headset Speech-to-Speech Translation System

Kristе Krstovski, Michael Decerbo, Rohit Prasad, David Stallard, Shirin Saleem, Premkumar Natarajan
Speech and Language Processing Department
BBN Technologies
10 Moulton Street, Cambridge, MA, 02138
{krstovski, mdecerbo, rprasad, stallard, ssaleem, prem}@bbn.com

Abstract

In this paper we present a wearable, headset integrated eyes- and hands-free speech-to-speech (S2S) translation system. The S2S system described here is configured for translingual communication between English and colloquial Iraqi Arabic. It employs an n-gram speech recognition engine, a rudimentary phrase-based translator for translating recognized Iraqi text, and a rudimentary text-to-speech (TTS) synthesis engine for playing back the English translation. This paper describes the system architecture, the functionality of its components, and the configurations of the speech recognition and machine translation engines.

1 Background

Humanitarian personnel, military personnel, and visitors in foreign countries often need to communicate with residents of a host country. Human interpreters are inevitably in short supply, and training personnel to speak a new language is difficult. Under the DARPA TRANSTAC and Babylon programs, various teams have developed systems that enable two-way communication over a language barrier (Waibel et al., 2003; Zhou et al., 2004; Stallard et al., 2006). The two-way speech-to-speech (S2S) translation systems seek, in principle, to translate any utterance, by using general statistical models trained on large amounts of speech and text data.

The performance and usability of such two-way speech-to-speech (S2S) translation systems is heavily dependent on the computational resources, such as processing power and memory, of the platform they are running on. To enable open-ended conversation these S2S systems employ powerful but highly memory- and computation-intensive statistical speech recognition and machine translation models. Thus, at the very minimum they require the processing and memory configuration of common-of-the-shelf (COTS) laptops.

Unfortunately, most laptops do not have a form factor that is suitable for mobile users. The size, weight, and shape of laptops render them unsuitable for handheld use. Moreover, simply carrying the laptop can be infeasible for users, such as military personnel, who are already overburdened with other equipment. Embedded platforms, on the other hand, offer a more suitable form factor in terms of size and weight, but lack the computational resources required to run more open-ended 2-way S2S systems.

In previous work, Prasad et al. (2007) reported on the development of a S2S system for Windows Mobile based handheld computers. To overcome the challenges posed by the limited resources of that platform, the PDA version of the S2S system was designed to be more constrained in terms of the ASR and MT vocabulary. As described in detail in (Prasad et al., 2007), the PDA based S2S system configured for English/Iraqi S2S translation delivers fairly accurate translation at faster than real-time.

In this paper, we present ongoing development work on an S2S system that runs on an even more constrained hardware platform; namely, a processor embedded in a wearable headset with just 32 MB of memory. Compared to the PDA based sys-

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 9–11,
Columbus, Ohio, USA, June 2008.

tem described in (Prasad et al., 2007), the wearable system is designed for both eyes- and hands-free operation. The headset-integrated translation device described in this paper is configured for two-way conversation in English/Iraqi. The target domain is the force protection, which includes scenarios of checkpoints, house searches, civil affairs, medical, etc.

In what follows, we discuss the hardware and software details of the headset-integrated translation device.

2 Hardware Platform

The wearable S2S system described in this paper runs on a headset-integrated computational platform developed by Integrated Wave Technologies, Inc. (IWT). The headset-integrated platform employs a 200 MHz StrongARM integer processor with a total of just 32MB RAM available for both the operating system and the translation software. The operating system currently running on the platform is Embedded Linux.

There are two audio cards on the headset platform for two-way communication through separate audio input and output channels. The default sound card uses the headset integrated close-talking microphone as an audio input and the second audio card can be used with an ambient microphone mounted on the device or an external microphone. In addition, each headset earpiece contains inner and outer set of speakers. The inner earpiece speakers are for the English speaking user who wears the headset, whereas the outer speakers are for the foreign language speaker who is not required to wear the headset.

3 Software Architecture

Depicted in Figure 1 is the software system architecture for the headset-integrated wearable S2S system. We are currently using a fixed-phrase English-to-Iraqi speech translation module from IWT for translating from English to Iraqi. In the Iraqi-to-English (I2E) direction, we use an n-gram ASR engine to recognize Iraqi speech, a custom, phrase-based "micro translator" for translating Iraqi text to English text, and finally a TTS module for converting the English text into speech. The rest of this paper focuses on the components of the Iraqi-to-English translation module.

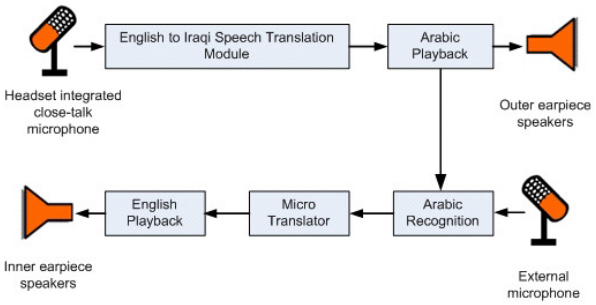


Figure 1. Software architecture of the S2S system.

Fixed point ASR Engine: The ASR engine uses phonetic hidden Markov models (HMM) with one or more forms of the following parameter tying: Phonetic-Tied Mixture (PTM), State-Tied Mixture (STM), and State-Clustered-Tied Mixture (SCTM) models.

For the headset-integrated platform, we use a fixed-point ASR engine described in (Prasad et al., 2007). As in (Prasad et al., 2007) for real-time performance we use the compact PTM models in both recognition passes of our two-pass ASR decoder.

Phrase-based Micro Translator: Phrase-based statistical machine translation (SMT) has been widely adopted as the translation engine in S2S systems. Such SMT engines require only a large corpus of bilingual sentence pairs to deliver robust performance on the domain of that corpus. However, phrase-based SMT engines require significant amount of memory, even when configured for medium vocabulary tasks. Given the limited memory on the headset platform, we chose to develop instead a phrase-based "micro translator" module, which acts like a bottom-up parser. The micro-translator uses translation rules derived from our phrase-based SMT engine. Rules are created automatically by running the SMT engine on a small training corpus and recording the phrase pairs it used in decoding it. These phrase pairs then become rules which are treated just as though they had been written by hand. The micro translator currently makes no use of probabilities. Instead, as shown in Figure 2, for any given Arabic utterance, the translator greedily chooses the longest matching source phrase that does not overlap a source phrase already chosen. The target phrases for these source phrases are then output as the translation. These target phrases come out in source-language

order, as no language model is currently used for reordering.

The micro translator currently consists of 1300 rules and 2000 words. Its memory footprint is just 32KB. This small memory footprint is achieved by representing the rules in binary format rather than text format.

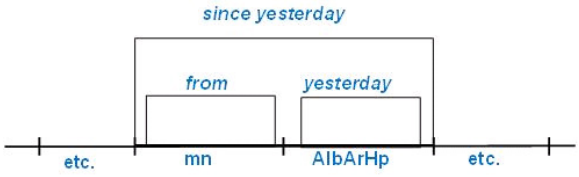


Figure 2. Decoding in micro translator.

English Playback using TTS: To play the English translation to the headset user we developed a rudimentary TTS module. The TTS module parses the output of the I2E translator to extract each translated word. It then uses the list of extracted words to read the appropriate pre-recorded (or synthesized) audio. Once the word pronunciations audio files are read we splice the beginning and the end of the audio files to reduce the amount of silence and concatenate them into a single file which is then played to the user on the inner earphone speakers.

The total memory footprint of our current Iraqi to English translation module running on the headset-integrated platform is just 9MB. The current configuration of the translation module's Iraqi ASR engine yields word error rate (WER) of 20% on test-set utterances without out-of-vocabulary (OOV) words.

4 Conclusions and Future Work

In this paper we have presented the initial setup of a speech-to-speech translation system configured for the headset platform. Our current work is focused on expanding the vocabulary of the Iraqi-to-English translation module by exploiting the rich morphology of Iraqi Arabic. In particular, we are investigating the use of morphemes (prefix, stems, and suffixes) for expanding the effective vocabulary of the headset translator. We are also developing use cases for performing a formal evaluation of both the usability and performance of the headset translator.

References

Alex Waibel, Ahmed Badran, Alan W Black, Robert Frederking, Donna Gates ,Alon Lavie, Lori Levin, Kevin Lenzo, Laura Mayfield Tomokiyo, J¨urgen Reichert, Tanja Schultz, Dorcas Wallace, Monika Woszczyna and Jing Zhang. 2003. "Speechalator: Two-way Speech-to-Speech Translation on a Consumer PDA," Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003), Geneva, Switzerland.

Bowen Zhou, Daniel D´echelotte and Yuqing Gao. 2004. "Two-way Speech-to-Speech Translation on Handheld Devices," Proc. 8th International Conference on Spoken Language Processing, Jeju Island, Korea.

David Stallard, Frederick Choi, Kriste Krstovski, Prem Natarajan and Shirin Saleem. 2006. "A Hybrid Phrase-based/Statistical Speech Translation System," Proc. The 9th International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP), Pittsburg, PA.

David Stallard, John Makhoul, Frederick Choi, Ehry Macrostie, Premkumar Natarajan, Richard Schwartz and Bushra Zawaydeh. 2003. "Design and Evaluation of a Limited two-way Speech Translator," Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003), Geneva, Switzerland.

Rohit Prasad, Kriste Krstovski, Frederick Choi, Shirin Saleem, Prem Natarajan, Michael Decerbo and David Stallard. 2007. "Real-Time Speech-to-Speech Translation for PDAs," Proc. IEEE International Conference on Portable Information Devices (IEEE Portable 2007), Orlando, FL.

Information extraction using finite state automata and syllable n-grams in a mobile environment

Choong-Nyoung Seon
Computer Science and Engineering
Sogang University
Seoul, Korea
wilowisp@gmail.com

Harksoo Kim
Computer and Communications Engineering
Kangwon National University
Chuncheon, Korea
nlpdrkim@kangwon.ac.kr

Jungyun Seo
Computer Science and Engineering
Sogang University
Seoul, Korea
seojy@sogang.ac.kr

Abstract

We propose an information extraction system that is designed for mobile devices with low hardware resources. The proposed system extracts temporal instances (dates and times) and named instances (locations and topics) from Korean short messages in an appointment management domain. To efficiently extract temporal instances with limited numbers of surface forms, the proposed system uses well-refined finite state automata. To effectively extract various surface forms of named instances with low hardware resources, the proposed system uses a modified HMM based on syllable *n*-grams. In the experiment on instance boundary labeling, the proposed system showed better performances than traditional classifiers.

1 Introduction

Recently, many people access various multi-media contents using mobile devices such as a cellular phone and a PDA (personal digital assistant). Accordingly, users' requests on NLP (natural language processing) are increasing because they want to easily and simply look up the multi-media contents. Information extraction is one of useful applications in NLP that helps users to easily access core information in a large amount of free texts. Unfortunately, it is not easy to implement an information extraction system in mobile devices because target texts include many morphological variations (*e.g.* blank omission, typos, word abbreviation) and mobile devices have many hardware limitations (*e.g.* a small volume of a main memory and the absence of an arithmetic logic unit for floating-point calculation)

There are some researches on information extraction from short messages in a mobile device, and Cooper's research (Cooper, 2005) is representative. Cooper predefined various syntactic patterns with placeholders and matched an input message against the syntactic patterns. Then, he extracted texts in the placeholders and assigned them the attribute name of the placeholders. This method has some advantages like easy implementation and fast response time. However, it is inadequate to apply Cooper's method to languages with partially-free word-order like Korean and Japanese because a huge amount of syntactic patterns should be predefined according as the degree of freedom on word order increases. Kang (2004) proposed a NLIDB (natural language interface to database) system using lightweight shallow NLP techniques. Kang raised problems of deep NLP techniques such as low portability and error-proneness. Kang proposed a lightweight approach to natural language interfaces, where translation knowledge is semi-automatically acquired and user questions are only syntactically analyzed. Although Kang's method showed good performances in spite of using shallow NLP techniques, it is difficult to apply his method to mobile devices because his method still needs a morphological analyzer with a large size of dictionary. In this paper, we propose an information extraction system that is designed for mobile devices with low hardware resources. The proposed system extracts appointment-related information (*i.e.* dates, times, locations, and topics) from Korean short messages.

This paper is organized as follows. In section 2, we proposed an information extraction system for a mobile device in an appointment domain. In sec-

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 12–17,
Columbus, Ohio, USA, June 2008.

tion 3, we explain experimental setup and report some experimental results. Finally, we draw some conclusions in section 4.

2 Lightweight information extraction system

Figure 1 shows an overall architecture of the proposed system.

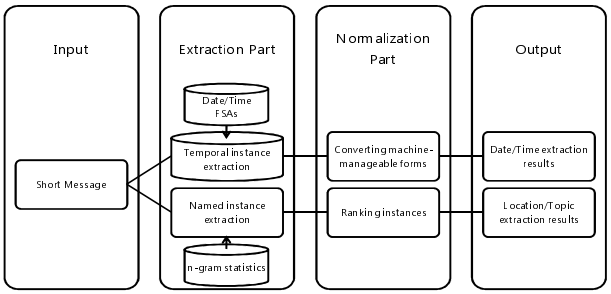


Figure 1. The system architecture

As shown in Figure 1, the proposed system consists of an extraction part and a normalization part. In the extraction part, the proposed system first extracts temporal instance candidates (*i.e.* dates and times) using FSA (finite-state automata). Then, the proposed system extracts named instance candidates (*i.e.* locations and topics) using syllable *n*-grams. Finally, the proposed system ranks the extracted instances and selects the highest one per target category. In the normalization part, the proposed system converts the temporal instances into suitable forms.

2.1 Information extraction using finite state automata

Although short messages in an appointment domain often include many incorrect words, temporal instances like dates and times are expressed as correct as possible because they are very important to appointment management. In addition, temporal instances are expressed in tractable numbers of surface forms in order to make message receivers easily be understood. In MUC-7, these kinds of temporal instances are called TIMEX (Chinchor, 1998), and it has known that TIMEX can be easily extracted by using FSA (Srihari, 2001). Based on these previous works, the proposed system extracts temporal instances from short messages by using FSA, as shown in Figure 2.

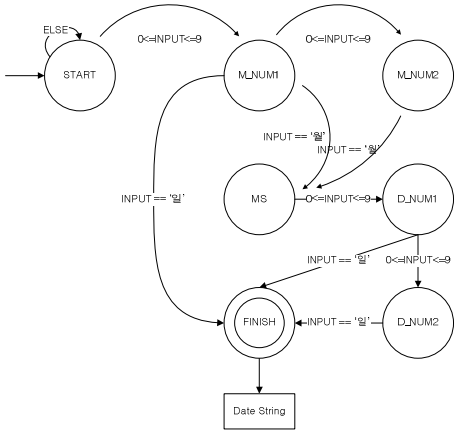


Figure 2. An example of FSA for date extraction

2.2 Information extraction using statistical syllable n-grams

Unlike dates and times, locations and topics not only have various surface forms, but also their constituent words are not included in a closed set. In MUC-7, these kinds of named entities are called NAMEX (Chinchor, 1998), and many researches on NAMEX have been performed by using rules and statistics. Generally, rule-based methods show high precisions but they have a weak point that it is hard to maintain a system when new words are continuously added to the system (Goh, 2003). Statistical methods guarantee reasonable performances but they need large-scale language resource and complex floating point operations. Therefore, it is not suitable to apply previous traditional approaches to mobile devices with many hardware limitations. To resolve this problem, we propose a statistical model based on syllable *n*-grams, as shown in Figure 3.

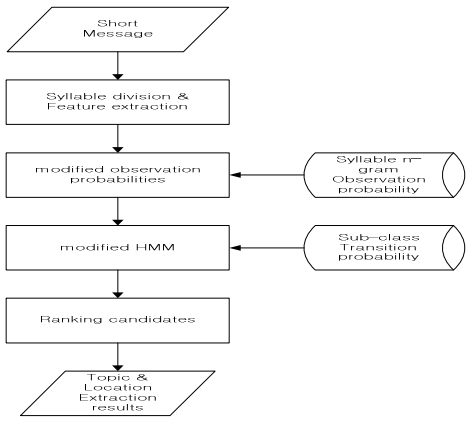


Figure 3. Statistical information extraction

The extraction of named instances has two kinds of problems; a instance boundary detection problem and a category assigning problem. If we can use a conventional morphological analyzer, the instance boundary detection problem is not big. However, it is not easy to use a morphological analyzer in a mobile device because of hardware limitations and users' writing habitations. Users often ignore word spacing and this habitation lowers the performance of the morphological analyzer. To resolve this problem, we adopt a syllable *n*-gram model that performs well in word boundary detection for languages like Chinese with no spacing between words (Goh, 2003; Ha, 2004). We first define 9 labels that represent boundaries of named instance candidates by adopting BIO (begin, inner, and outer) annotation scheme, as shown in Table 1 (Hong, 2005; Uchimoto, 2000).

Tag	Description	Tag	Description
LB	Begin of a location	TB	Begin of a topic
LI	Inner of a location	TI	Inner of a topic
LE	End of a location	TE	End of a topic
LS	A single-syllable location	TS	A single-syllable topic
OT	Other syllable		

Table 1. The definition of instance boundary labels

Then, based on a modified HMM (hidden Markov model), the proposed system assigns boundary labels to each syllable in an input message, as follows.

Let $S_{1,n}$ denote a message which consists of a sequence of *n* syllable, $s_1, s_2, ..., s_n$, and let $L_{1,n}$ denote the boundary label sequence, $l_1, l_2, ..., l_n$, of $S_{1,n}$. Then, the label annotation problem can be formally defined as finding $L_{1,n}$ which is the result of Equation (1).

$$\begin{aligned} L(S_{1,n}) &\overset{def}{=} \arg\max_{L_{1,n}} P(L_{1,n} \mid S_{1,n}) \\ &= \arg\max_{L_{1,n}} \frac{P(L_{1,n}, S_{1,n})}{P(S_{1,n})} \\ &= \arg\max_{L_{1,n}} P(L_{1,n}, S_{1,n}) \end{aligned} \quad (1)$$

In Equation (1), we dropped $P(S_{1,n})$ as it is constant for all $L_{1,n}$. Next, we break Equation (1) into bite-size pieces about which we can collect statistics, as shown in Equation (2).

$$P(L_{1,n}, S_{1,n}) = \prod_{i=1}^{n} P(s_i \mid l_{1,i}, s_{1,i-1}) P(l_i \mid l_{1,i-1}, s_{1,i-1}) \quad (2)$$

We simplify Equation (2) by making the following two assumptions: one is that the current boundary label is only dependent on the previous boundary label, and the other is that current boundary label is affected by its contextual features.

$$P(L_{1,n}, S_{1,n}) = \prod_{i=1}^{n} P^*(s_i \mid l_i) P(l_i \mid l_{i-1}) \quad (3)$$

In Equation (3), $P^*(s_i \mid l_i)$ is a modified observation probability that is adopted from a class probability in naïve Bayesian classification (Zheng, 1998) as shown in Equation (4). The reason why we modify an original observation probability $P(s_i \mid l_i)$ in HMM is its sparseness that is caused by a size limitation of training corpus in a mobile environment.

$$P^*(s_i \mid l_i) = \frac{1}{Z} P(l_i) \prod_{j=1}^{f} P(s_{ij} \mid l_i) \quad (4)$$

In Equation (4), f is the number of contextual features, and s_{ij} is the *j*th feature of the *i*th syllable. Z is a normalizing factor. Table 2 shows the contextual features that the proposed system uses.

Feature	Composition	Meaning
s_{i1}	s_i	The current syllable
s_{i2}	$s_{i-1}s_i$	The previous syllable and the current syllable
s_{i3}	$s_i s_{i+1}$	The current syllable and the next syllable

Table 2. The composition of contextual features

In Equation (1), the max scores are calculated by using the well-known Viterbi algorithm (Forney, 1973).

After performing instance boundary labeling, the proposed system extracts syllable sequences labeled with the same named categories. For example, if a syllable sequence is labeled with 'TS OT LB LI LI', the proposed system extracts the sub-sequence of syllables labeled with 'LB LI LI',

as a location candidate. Then, the proposed system ranks the extracted instance candidates by using some information such as position, length, and a degree of completion, as shown in Equation (5).

$$Rank(NI_i) = \alpha \cdot Position_i + \beta \cdot Length_i + \gamma \cdot Completion_i \quad (5)$$

In Equation (5), $Position_i$ means the distance from the beginning of input message to the *i*th named instance candidate NI_i. In Korean, important words tend to appear in the latter part of a message. Therefore, we assume that the latter part an instance candidate appears in, the more important the instance candidate is. $Length_i$ means the length of an instance candidate. We assume that the longer an instance candidate include is, the more informative the instance candidate is. $Completion_i$ means whether a sequence of instance boundary labels is complete. We assume that instance candidates with complete label sequences are more informative. To check the degree of completion, the proposed system uses FSA, as shown in Figure 4. In the training corpus, every transition is legal. Therefore most of candidates were satisfied the completion condition. However, sometimes the completion condition is not satisfied, when the candidate was extracted from the boundary of a sentence. Accordingly the condition gave an effect to the rank.

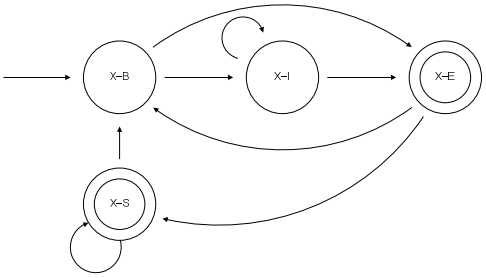


Figure 4. The FSA for checking label completion

In the experiments, we set α, β, and γ to 1, 2, and 10, respectively.

2.3 Normalization of temporal instances

It is inadequate for the proposed system to use the extracted temporal instances as database instances without any processing because the temporal instances consist of various forms of human-readable strings like 'January 24, 2008'. Therefore, the proposed system should normalize the temporal instances into machine-manageable forms like '20080124'. However, the normalization is not easy because temporal instances often include the relative information like 'this Sunday' and 'after two days'. To resolve this problem, the proposed system converts relative temporal instances into absolute temporal instances by using a message arrival time. For example, if a message includes the temporal instance 'after two days', the proposed system checks arrival time information of the message. Then, the proposed system adds a date in the arrival time information to two days. Figure 5 shows an example of date normalization.

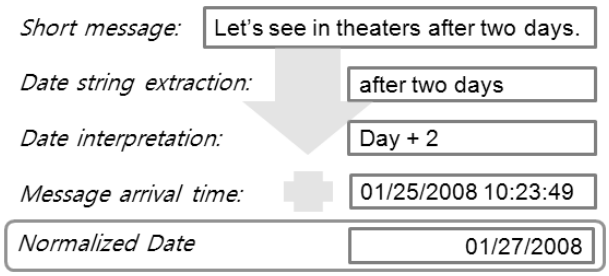


Figure 5. An example of date normalization

3 Evaluation

3.1 Data sets and experimental settings

We collected 6,190 short messages simulated in an appointment scheduling domain. These messages contain 4,686 locations and 4,836 topics. Each message is manually annotated with the boundary labels in Table 1. The manual annotation was done by 2 graduate students majoring in natural language processing and post-processed by a student in a doctoral course for consistency. In order to experiment the proposed system, we divided the annotated messages into the training corpus and the testing corpus by a ratio of four (4,952 messages) to one (1,238 messages). Then, we performed 5-fold cross validation and used a precision, a recall rate, and a F1-measure as performance measures. In this paper, we did not evaluate the performances on the temporal instance extraction because performances of the proposed method are fully dependent on the coverage of pre-constructed FSA.

3.2 Experimental results

To choose the proper size of language models in a mobile environment, we evaluated performance variations of the proposed system, as shown in Figure 6.

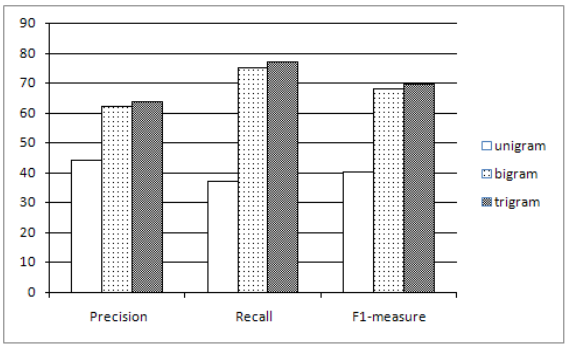


Figure 6. The performance variations according to the size of language models

As shown in Figure 6, the system using syllable unigrams showed much lower performances than the systems using syllable bigrams or syllable trigrams.

	Bigram	Trigram
# of features	54,711	158,525
Size of DB	1.33M	2.83M

Table 3. Space requirements of language models.

However, as shown in the Table 3, although the number of syllable trigrams was three times larger than the number of syllable bigrams, the difference of performances between the system using syllable bigrams and the system using syllable trigrams was not big (about 1%). Based on this experimental result, we conclude that the combination of syllable unigrams and syllable bigrams, as shown in Table 2, is the most suitable language model for mobile devices with low hardware resources.

To evaluate the proposed system, we calculated two types of performances. One is boundary labeling performances that measure whether the proposed system can correctly annotate a test corpus with boundary labels in Table 1. The other is extraction performances that measure whether the proposed system can correctly extract named instances from a test corpus by using Equation (5). Table 4 shows the boundary labeling performances of the proposed system in comparisons with those of representative classifiers.

Model	Precision	Recall rate	F1-measure
NB	62.74%	75.17%	68.34%
SVM	67.29%	67.58%	67.37%
CRF	70.98%	66.27%	68.45%
Proposed system	74.81%	77.20%	75.91%

Table 4. The comparison of boundary labeling performances

In Table 4, NB is a classifier using naïve Bayesian statistics, and SVM is a classifier using a support vector machine. CRF is a classifier using conditional random fields. As shown in Table 4, the proposed system outperformed the comparative models in all measures. Based on this fact, we think that the modified HMM may be more effective in a labeling sequence problem.

Table 5 shows the extraction performances of the proposed system. In Table 5, the reason why the performances on the topic extraction are lower is that topic instances can consist of more various syllables (e.g. the topic instance, 'a meeting in Samsung Research Center', includes the location, 'Samsung Research Center').

Category	Precision	Recall rate	F1-measure
Location	79.37%	76.33%	77.78%
Topic	58.54%	55.20%	56.72%

Table 5. The extraction performances

Table 6 shows performance variations according as the parameters in Equation (5) are changed. As shown in Table 6, the differences between performances are not big, and the proposed model showed the best performance at (α=1, β=2, γ=5) or (α=1, β=2, γ=10). On the basis of this experiments, we set α, β, and γ to 1, 2, and 5, respectively.

(α,β,γ)	Precision of Location	Recall rate of Location	F1-measure of Location
(1,1,1)	79.23%	76.20%	77.65%
(1,1,5)	79.28%	76.24%	77.69%

(1,1,10)	79.30%	76.26%	77.71%
(1,2,5)	**79.37%**	**76.33%**	**77.78%**
(1,2,10)	**79.37%**	**76.33%**	**77.78%**
(α,β,γ)	Precision of Topic	Recall rate of Topic	F1-measure of Topic
(1,1,1)	58.09%	54.76%	56.28%
(1,1,5)	58.09%	54.76%	56.28%
(1,1,10)	58.11%	54.78%	56.30%
(1,2,5)	**58.54%**	**55.20%**	**56.72%**
(1,2,10)	**58.54%**	**55.20%**	**56.72%**

Table 6. The performance variations according to parameter changes

To evaluate usefulness of the proposed model in a real mobile phone environment, we measured an average response time of 100 short messages in a mobile phone with XSCALE PXA270 CPU, 51.26MB memory, and Windows mobile 5.0. We obtained an average response time of 0.0532 seconds.

4 Conclusion

We proposed an information extraction system for a mobile device in an appointment management domain. The proposed system efficiently extracts temporal instances with limited numbers of surface forms by using FSA. To effectively extract various surface forms of named instances with low hardware resources, the proposed system uses a modified HMM based on syllable n-grams. In the experiment on instance boundary labeling, the proposed system outperformed traditional classifiers that showed good performances in a labeling sequence problem. On the experimental basis, we think that the proposed method is very suitable for information extraction applications with many hardware limitations.

Acknowledgments

This research (paper) was funded by Samsung Electronics.

5 Reference

Chooi Ling Goh, Masayuki Asahara, Yuji Matsumoto. 2003. Chinese unknown word identification using character-based tagging and chunking. *Proceedings of ACL-2003 Interactive Posters and Demonstrations*, 197-200.

G. David Forney, JR. 1973. The Viterbi Algorithm *Proceedings of the IEEE,* 61(3):268-278.

Hong Shen, Anoop Sarkar. 2005. Voting Between Multiple Data Representations for Text Chunking. *Canadian Conference on AI 2005*. 389-400.

In-Su Kang, Seung-Hoon Na, Jong-Hyeok Lee, Gijoo Yang. 2004. Lightweight Natural Language Database Interfaces. *Proceedings of the 9th International Conference on Application of Natural Language to Information Systems*. 76-88.

Juhong Ha, Yu Zheng, Byeongchang Kim, Gary Geunbae Lee, Yoon-Suk Seong. 2004. High Speed Unknown Word Prediction Using Support Vector Machine for Chinese Text-to-Speech Systems. IJCNLP:509-517

Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In Proceedings of the 38th Annual Meeting of Association for Computational Linguistics

Nancy A. Chinchor. 1998. MUC-7 named entity task definition, Proceedings of the Seventh Message Understanding Conference.

Richard Cooper, Sajjad Ali, Chenlan Bi, 2005. Extracting Information from Short Messages, NLDB 2005, LNCS 3513, pp. 388-391.

Rohini Srihari, Cheng Niu, Wei Li. 2001. A hybrid approach for named entity and sub-type tagging. In Proc. 6th Applied Natural Language Processing Conference.

Zijian Zheng. Naive Bayesian classifier committees. Proceedings of the 10th European Conference on Machine Learning. Berlin: Springet-Verlag (1998) 196-207.

SVM_light: http://svmlight.joachims.org/

CRF++: http://crfpp.sourceforge.net/

Small Statistical Models by Random Feature Mixing

Kuzman Ganchev and **Mark Dredze**
Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA
{kuzman,mdredze}@cis.upenn.edu

Abstract

The application of statistical NLP systems to resource constrained devices is limited by the need to maintain parameters for a large number of features and an alphabet mapping features to parameters. We introduce random feature mixing to eliminate alphabet storage and reduce the number of parameters without severely impacting model performance.

1 Introduction

Statistical NLP learning systems are used for many applications but have large memory requirements, a serious problem for mobile platforms. Since NLP applications use high dimensional models, a large alphabet is required to map between features and model parameters. Practically, this means storing every observed feature string in memory, a prohibitive cost for systems with constrained resources. Offline feature selection is a possible solution, but still requires an alphabet and eliminates the potential for learning new features after deployment, an important property for adaptive e-mail or SMS prediction and personalization tasks.

We propose a simple and effective approach to *eliminate* the alphabet and reduce the problem of dimensionality through random feature mixing. We explore this method on a variety of popular datasets and classification algorithms. In addition to alphabet elimination, this reduces model size by a factor of 5–10 without a significant loss in performance.

2 Method

Linear models learn a weight vector over features constructed from the input. Features are constructed as strings (e.g. "`w=apple`" interpreted as "contains the word apple") and converted to feature indices maintained by an alphabet, a map from strings to integers. Instances are efficiently represented as a sparse vector and the model as a dense weight vector. Since the alphabet stores a string for each feature, potentially each unigram or bigram it encounters, it is much larger than the weight vector.

Our idea is to replace the alphabet with a random function from strings to integers between 0 and an intended size. This size controls the number of parameters in our model. While features are now easily mapped to model parameters, multiple features can collide and confuse learning. The collision rate is controlled by the intended size. Excessive collisions can make the learning problem more difficult, but we show significant reductions are still possible without harming learning. We emphasize that even when using an extremely large feature space to avoid collisions, alphabet storage is eliminated. For the experiments in this paper we use Java's `hashCode` function modulo the intended size rather than a random function.

3 Experiments

We evaluated the effect of random feature mixing on four popular learning methods: Perceptron, MIRA (Crammer et al., 2006), SVM and Maximum entropy; with 4 NLP datasets: 20 Newsgroups[1], Reuters (Lewis et al., 2004), Sentiment (Blitzer et al., 2007) and Spam (Bickel, 2006). For each dataset we extracted binary unigram features and sentiment was prepared according to Blitzer et al. (2007). From 20 Newsgroups we created 3 binary decision tasks to differentiate between two similar

[1] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 18–19,
Columbus, Ohio, USA, June 2008.

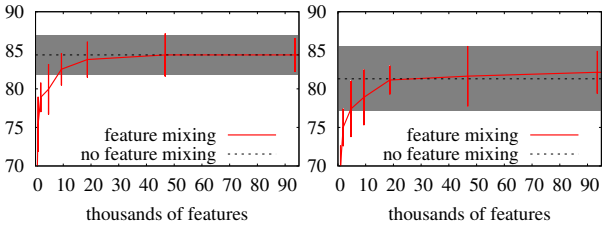


Figure 1: Kitchen appliance reviews. Left: Maximum entropy. Right: Perceptron. Shaded area and vertical lines extend one standard deviation from the mean.

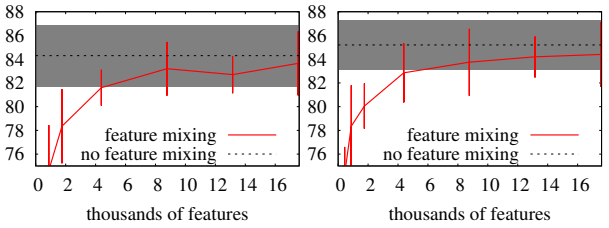


Figure 3: The anomalous Reuters dataset from figure 2 for Perceptron (left) and MIRA (right).

labels from computers, science and talk. We created 3 similar problems from Reuters from insurance, business services and retail distribution. Sentiment used 4 Amazon domains (book, dvd, electronics, kitchen). Spam used the three users from task A data. Each problem had 2000 instances except for 20 Newsgroups, which used between 1850 and 1971 instances. This created 13 binary classification problems across four tasks. Each model was evaluated on all problems using 10-fold cross validation and parameter optimization. Experiments varied model size to observe the effect of feature collisions on performance.

Results for sentiment classification of kitchen appliance reviews (figure 1) are typical. The original model has roughly 93.6k features and its alphabet requires 1.3MB of storage. Assuming 4-byte floating point numbers the weight vector needs under 0.37MB. Consequently our method reduces storage by over 78% when we keep the number of parameters constant. A further reduction by a factor of 2 decreases accuracy by only 2%.

Figure 2 shows the results of all experiments for SVM and MIRA. Each curve shows normalized dataset performance relative to the full model as the percentage of original features decrease. The shaded rectangle extends one standard deviation above and below full model performance. Almost all datasets perform within one standard deviation of the full model when using feature mixing set to the total number of features for the problem, indicating that alphabet elimination is possible without hurting performance. One dataset (Reuters retail distribution) is a notable exception and is illustrated in detail in figure 3. We believe the small total number of features used for this problem is the source of this behavior. On the vast majority of datasets, our method can reduce the size of the weight vector and eliminate the alphabet without any feature selection or changes to the learning algorithm. When reducing weight vector size by a factor of 10, we still obtain between 96.7% and 97.4% of the performance of the original model, depending on the learning algorithm. If we eliminate the alphabet but keep the same size weight vector, model the performance is between 99.3% of the original for MIRA and a slight improvement for Perceptron. The batch learning methods are between those two extremes at 99.4 and 99.5 for maximum entropy and SVM respectively. Feature mixing yields substantial reductions in memory requirements with a minimal performance loss, a promising result for resource constrained devices.

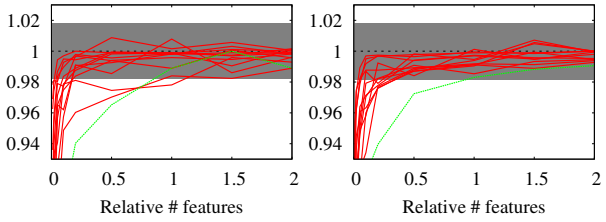


Figure 2: Relative performance on all datasets for SVM (left) and MIRA (right).

References

S. Bickel. 2006. Ecml-pkdd discovery challenge overview. In *The Discovery Challenge Workshop*.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Ressearch*, 7.

D. D. Lewis, Y. Yand, T. Rose, and F. Li. 2004. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397.

Mixture Pruning and Roughening for Scalable Acoustic Models

David Huggins-Daines
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dhuggins@cs.cmu.edu

Alexander I. Rudnicky
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
air@cs.cmu.edu

Abstract

In an automatic speech recognition system using a tied-mixture acoustic model, the main cost in CPU time and memory lies not in the evaluation and storage of Gaussians themselves but rather in evaluating the mixture likelihoods for each state output distribution. Using a simple entropy-based technique for pruning the mixture weight distributions, we can achieve a significant speedup in recognition for a 5000-word vocabulary with a negligible increase in word error rate. This allows us to achieve real-time connected-word dictation on an ARM-based mobile device.

1 Introduction

As transistors shrink and CPUs become faster and more power-efficient, we find ourselves entering a new age of intelligent mobile devices. We believe that not only will these devices provide access to rich sources of on-line information and entertainment, but they themselves will find new applications as personal knowledge management agents. Given the constraints of the mobile form factor, natural speech input is crucial to these applications. However, despite the advances in processor technology, mobile devices are still highly constrained by their memory and storage subsystems.

2 Semi-Continuous Acoustic Models

Recent research into acoustic model compression and optimization of acoustic scoring has focused on "Fully Continuous" acoustic models, where each Hidden Markov Model state's output probability distribution is modeled by a mixture of multivariate Gaussian densities. This type of model allows large amounts of training data to be efficiently exploited to produce detailed models. However, due to the large number of parameters in these models, approximate computation techniques (Woszczyna, 1998) are required in order to achieve real-time recognition even on workstation-class hardware.

Another historically popular type of acoustic model is the so-called "Semi-Continuous" or tied-mixture model, where a single codebook of Gaussians is shared by all HMM states (Huang, 1989). The parameters of this codebook are updated using the usual Baum-Welch equations during training, using sufficient statistics from all states. The mixture weight distributions therefore become the main source of information used to distinguish between different speech sounds.

There are several benefits to semi-continuous models for efficient speech recognition. The most obvious is the greatly reduced number of Gaussian densities which need to be computed. With fully continuous models, we must compute one codebook of 16 or more Gaussians for each HMM state, of which there may be several thousand active for any given frame of speech input. For semi-continuous models, there is a single codebook with a small number of Gaussians, typically 256. In addition, since only a few Gaussians will have non-negligible densities for each frame of speech, and this set of Gaussians tends to change slowly, partial computation of each density is possible.

Another useful property of semi-continuous models is that the mixture weights for each state have the form of a multinomial distribution, and are thus amenable to various smoothing and adaptation techniques. In particular, Bayesian and quasi-Bayes

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 20–23,
Columbus, Ohio, USA, June 2008.

adaptation (Huo and Chan, 1995) are effective and computationally efficient.

3 Experimental Setup

All experiments in this paper were performed using PocketSphinx (Huggins-Daines et al., 2006). The baseline acoustic model was trained from the combined WSJ0 and WSJ1 "long" training sets (Paul and Baker, 1992), for a total of 192 hours of speech. This speech was converted to MFCC features using a bank of 20 mel-scale filters spaced from 0 to 4000Hz, allowing the model to work with audio sampled at 8kHz, as is typical on mobile devices. We used 5-state Hidden Markov Models for all phones. Output distributions were modeled by a codebook of 256 Gaussians, shared between 5000 tied states and 220 context-independent states. Only the first pass of recognition (static lexicon tree search) was performed.

Our test platform is the Nokia N800, a handheld Internet Tablet. It uses a Texas Instruments OMAP^{TM} 2420 processor, which combines an ARM11 RISC core and a C55x DSP core on a single chip. The RISC core is clocked at 400MHz while the DSP is clocked at 220MHz. In these experiments, we used the ARM core for all processing, although we have also ported the MFCC extraction code to the DSP. The decoder binaries, models and audio files were stored on a high-speed SD flash card formatted with the `ext3` journaling filesystem. Using the standard `bcb05cnp` bigram language model, we obtained a baseline word error rate of 9.46% on the `si_et_05` test set. The baseline performance of this platform on the test set is 1.40 times real-time, that is, for every second of speech, 1.40 seconds of CPU time are required for recognition.

We used the `oprofile` utility[1] on the Nokia N800 to collect statistical profiling information for a subset of the test corpus. The results are shown in Table 1. We can see that three operations occupy the vast majority of CPU time used in decoding: managing the list of active HMM states, computing the codebook of Gaussians, and computing mixture densities.

The size of the files in the acoustic model is shown in Table 2. The amount of CPU time required to calculate densities is related to the size of the mixture weight distribution by the fact that the N800 has a single-level 32Kb data cache, while a typical desktop processor has two levels of cache totalling at least 1Mb. We used `cachegrind`[2] to simulate the memory hierarchy of an OMAP versus an AMD K8 desktop processor with 64Kb of L1 cache and 512Kb of L2 cache, with results shown in Table 3.

While other work on efficient recognition has focused on quantization of the Gaussian parameters (Leppänen and Kiss, 2005), in a semi-continuous model, the number of these parameters is small

Function	%CPU
HMM evaluation	**22.41**
hmm_vit_eval_5st_lr	13.36
hmm_vit_eval_5st_lr_mpx	3.71
Mixture Evaluation	**21.66**
get_scores4_8b	14.94
fast_logmath_add	6.72
Lexicon Tree Search	**19.89**
last_phone_transition	5.25
prune_nonroot_chan	4.15
Active List Management	**15.57**
hmm_sen_active	13.75
compute_sen_active	1.19
Language Model Evaluation	**7.80**
find_bg	2.55
ngram_ng_score	2.13
Gaussian Evaluation	**5.87**
eval_cb	5.59
eval_topn	0.28
Acoustic Feature Extraction	**3.60**
fe_fft_real	1.59
fixlog2	0.77

Table 1: CPU profiling, OMAP platform

File	Size (bytes)
`sendump` (mixture weights)	5345920
`mdef` (triphone mappings)	1693280
`means` (Gaussians)	52304
`variances` (Gaussians)	52304
`transition_matrices`	5344

Table 2: File sizes, WSJ1 acoustic model

[1] `http://oprofile.sourceforge.net/`

[2] `http://valgrind.org/`

Function	ARM	K8
hmm_vit_eval_5st_lr	4.71	3.95
hmm_sen_active	3.55	3.76
get_scores4_8b	2.87	1.92
prune_root_chan	2.07	2.29
prune_nonroot_chan	1.99	1.73
eval_cb	1.73	1.77
hmm_vit_eval_5st_lr_mpx	1.30	0.80

Table 3: Data cache misses (units of 10^7)

enough that little cost is incurred by storing and calculating them as 32-bit fixed-point numbers. Therefore, we focus here on ways to reduce the amount of storage and computation used by the mixture weight distributions.

4 Mixture Roughening

Our method for speeding up mixture computation is based on the observation that mixture weight distributions are typically fairly "spiky", with most of the probability mass concentrated in a small number of mixture weights. One can quantify this by calculating the perplexity of the mixture distributions:

$$pplx(w_i) = \exp \sum_{k=0}^{N} w_{ik} \log \frac{1}{w_{ik}}$$

A histogram of perplexities is shown in Figure 1. The perplexity can be interpreted as the average number of Gaussians which were used to generate an observation drawn from a particular distribution. Therefore, on average, the vast majority of the 256 Gaussians contribute minimally to the likelihood of the data given a particular mixture model.

When evaluating mixture densities with pruned models, one can either treat these mixture weights as having a small but non-negligible value, or set them to zero[3]. Note that the mixture weights are renormalized in both cases, and thus the former is more or less equivalent to add-one smoothing. The latter can be thought of as exactly the opposite of smoothing - "roughening" the distribution. To investigate this, we set all but the top 16 values in each mixture weight distribution to zero and ran a number of trials on a K8-based workstation, varing the

[3]Meaning a very small number, since they are stored in log domain.

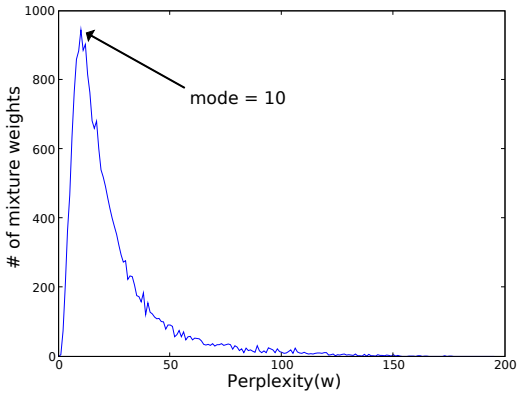


Figure 1: Perplexity distribution of mixture weights

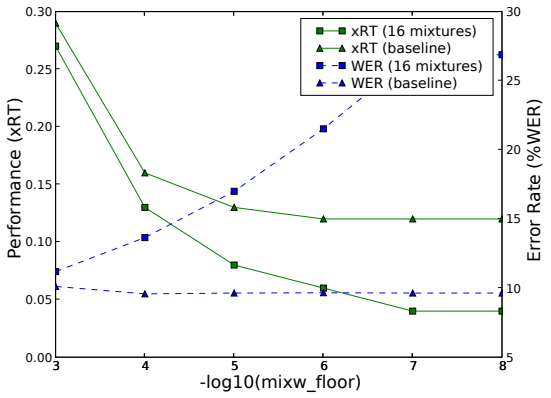


Figure 2: Smoothing vs. Roughening, 16 mixtures

mixture weight floor to produce either a smoothing or roughening effect. We discovered something initially surprising: "roughening" the mixture weights speeds up decoding significantly, while smoothing them slows it down. A plot of speed and error rate versus mixture weight floor is shown in Figure 2.

However, there is a simple explanation for this. At each frame, only the top N Gaussian densities are actually used to calculate the likelihood of the data:

$$p(x|\lambda_i) = \sum_{k \in topN} w_{ik} N(x; \vec{\mu}_{ik}, \vec{\sigma}^2_{ik})$$

When we remove mixture weights, we increase the probability that these top N densities will be matched with pruned-out weights. If we smooth the weights, we may raise some of these weights above their maximum-likelihood estimate, thus increasing

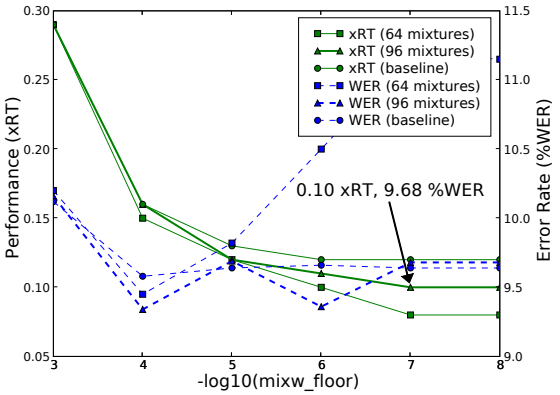


Figure 3: Speed-accuracy tradeoff with pruned mixtures

the likelihood for models whose top mixture weights do not overlap with the top N densities. This may prevent HMM states whose output distributions are modeled by said models from being pruned by beam search, therefore slowing down the decoder. By "roughening" the weights, we decrease the likelihood of the data for these models, and hence make them more likely to be pruned, speeding up the decoder and increasing the error rate. This is a kind of "soft" GMM selection, where instead of excluding some models, we simply make some more likely and others less likely.

As we increase the number of retained mixture weights, we can achieve an optimal tradeoff between speed and accuracy, as shown in Figure 3. Additionally, the perplexity calculation suggests a principled way to vary the number of retained mixture weights for each model. We propose setting a target number of mixture weights, then calculating a scaling factor based on the ratio of this target to the average perplexity of all models:

$$topK_i = \frac{target}{\frac{1}{N}\sum_{i=0}^{N} pplx(w_i)} pplx(w_i)$$

One problem is that many models have very low perplexity, such that we end up retaining only a few mixture weights. When the mixture weights are "roughened", this guarantees that these models will score poorly, regardless of the data. We compensate for this by keeping a minimum number of mixture weights regardless of the perplexity. Using a target of 96 mixtures, a minimum of 16, and a mixture weight floor of 10^{-8}, we achieve 9.90% word error rate in 0.09 times real-time, a 21% speedup with a 2.7% relative increase in error (baseline error rate is 9.64% on the desktop).

Using the same entropy-pruned mixture weights on the N800, we achieve an error rate of 9.79%, running in 1.19 times real-time, a 15% speedup with a 3.4% relative increase in error. After applying absolute pruning thresholds of 800 HMMs per frame and 5 words per frame, we obtained a 10.01% word error rate in 1.01 times real-time.

5 Conclusion

We have shown that a simple pruning technique allows acoustic models trained for large-vocabulary continuous speech recognition to be "scaled down" to run in real-time on a mobile device without major increases in error. In related work, we are experimenting with bottom-up clustering techniques on the mixture weights to produce truly scalable acoustic models, and subvector clustering to derive semi-continuous models automatically from well-trained fully-continuous models.

Acknowledgments

We wish to thank Nokia for donating the N800 tablet used in these experiments.

References

X. D. Huang. 1989. *Semi-continuous Hidden Markov Models for Speech Recognition*. Ph.D. thesis, University of Edinburgh.

D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky. 2006. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of ICASSP 2006*, Toulouse, France.

Q. Huo and C. Chan. 1995. On-line Bayes adaptation of SCHMM parameters for speech recognition. In *Proceedings of ICASSP 1995*, Detroit, USA.

J. Leppänen and I. Kiss. 2005. Comparison of low footprint acoustic modeling techniques for embedded ASR systems. In *Proceedings of Interspeech 2005*, Lisbon, Portugal.

D. Paul and J. Baker. 1992. The design for the Wall Street Journal based CSR corpus. In *Proceedings of the ACL workshop on Speech and Natural Language*.

M. Woszczyna. 1998. *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, University of Karlsruhe.

Assistive Mobile Communication Support

Sonya Nikolova and Xiaojuan Ma
Princeton University, Princeton, NJ 08540, USA
{nikolova, xm}@cs.princeton.edu

Abstract

This paper reflects on our work in providing communication support for people with speech and language disabilities. We discuss the role of mobile technologies in assistive systems and share ongoing research efforts.

1 Introduction

Designing assistive communication technologies for people with speech and language disabilities involves a number of challenges including addressing stigma and portability concerns. Mobile devices have the potential to resolve some of these issues if disadvantages such as small screen size and difficult interaction techniques are tackled.

Our experience shows that a mobile device and a desktop computer can serve in synergy as an effective communication support for people with aphasia, a disability that impairs the language modalities. Our work has also showed that assisting the user effectively is interlaced with the ability to provide flexible and personalizable support.

2 Mobile Support for Communication

Being able to communicate is essential to leading a self-sufficient and satisfying life. Individuals who suffer from aphasia experience many challenges, including social isolation (Kauhanen et al., 2000). Technology has the potential to help such individuals, but to enhance the users' daily communication effectively, tools need to be portable and usable outside of the home.

Our research focuses on designing multimodal communication systems for people with aphasia. We are interested in small and light-weight devices because their use is inconspicuous in public which addresses stigma issues prevalent among users with disabilities. Mobile devices have obvious shortcomings such as limited screen size and interaction techniques, and relatively small memory. However, if they assume the role of an extension instead of a replacement of traditional desktop and laptop assistive systems, we can easily take advantage of their positive characteristics–portability being the most important one. In addition, mobile devices which have embedded camera, microphone, and speaker are convenient for taking photos, recording videos and sounds which can enhance communication. To compensate for the loss of language, visual and audio representations are essential support for information comprehension and speech production.

There has been consistently increasing interest in using mobile platforms for applications that support communication. For example, Davies et al. (2004) extensively studied one individual with aphasia who incorporated a personal digital assistant (PDA) into his daily communication strategies and demonstrated the device's potential. Moffatt et al. (2004) implemented an electronic daily planner enhanced with images and sounds, running on a PDA. Even though it received positive feedback when evaluated with aphasic individuals, the prevalently elderly subjects had difficulty in composing and editing appointments directly on the PDA.

Considering the advantages and disadvantages of a personal digital assistant, we combined a PDA and a desktop computer into a hybrid communication system for people with aphasia.

3 Hybrid Communication System

The desktop component of the hybrid communication system is used to compose and edit sentences and the PDA is used as a portable extension for conversations outside of home. A multimodal approach is used to compensate for the considerable variability in language impairments among individuals with aphasia. In the system's vocabulary, nouns are represented by text, sound and an image while verbs are represented by text, sound and an animation depicting the action. Users enjoyed and were able to work with the system, and they incorporated multiple photographs taken with the PDA into their communication (Boyd-Graber et al., 2006). The evaluation also revealed certain weaknesses and confirmed the need for flexibility and

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 24–25,
Columbus, Ohio, USA, June 2008.

customization in assistive technology outlined in previous work (Moffatt et al., 2004 and van de Sandt-Koenderman et al., 2005).

4 Work-in-Progress

We are currently redesigning the system and improving its flexibility by introducing some adaptive and adaptable features.

4.1 Web-based System

The fact that most mobile devices nowadays can access and browse the Internet relatively easily encourages us to explore a web-based system. By having the desktop and mobile components communicate online, we circumvent the time and location constraints of traditional synchronization methods. Even though data can be transferred easily between the two components of our system, the need for physical contact is a drawback if multiple parties are involved in the information sharing (for example, when the aphasia patient, her speech-language pathologist and her caregiver need to access and modify information on different desktops at different locations).

A web-based system will also eliminate the dependence on the mobile device's limited storage, which dictates the quantity and quality of the multimedia data available to the user. The user will have access to more pictures, videos, and audio clips stored on the server. A web-based system will also allow for sharing of resources online among users within the aphasic community which could forge new social connections.

4.2 Building an Adaptable Vocabulary

An essential component of a communication system that attempts to be flexible, extensible, and expressive is the vocabulary that it offers to the users. Vocabulary depth, breadth, organization and management are major challenges in existing assistive technologies including our own.

We are addressing these problems by designing a vocabulary application enhanced with adaptive and adaptable features. The goal is to enable aphasic users to build phrases quickly by browsing through a smart network of words represented by a triplet of image, sound and text. The links between words are based on the individual's vocabulary profile (frequently used words, personal interests and communication context) as well as word similarity and evocation measures derived from WordNet (Miller, 1990). The system's adaptability will let the user add and remove words, group the words in a personalized manner by creating their own categories (such as a "Favorites" folder) and enhance them with images and sounds. The adaptive part will study the usage frequency of each word, make context relevant suggestions, and adapt the vocabulary organization automatically. Thus, frequently used words and words relevant to the user's interests or the context of the communication will surface faster. Naturally, the challenge is balancing the adaptive and adaptable aspects of the system to best benefit the user.

5 Conclusion

Even though small screen size and challenging interaction techniques make most existing mobile devices unusable for persons with aphasia, they hold a significant potential to assist daily communication effectively. We envision that a web-based communication system with a mobile and a desktop component will be an effective support because it will eliminate constraints related to data transfer and information updating. We are also working on creating adaptive and adaptable techniques that allow the vocabulary used for communication to be tailored to the user's needs. We are interested in discussing effective design strategies and interaction technique for language application for mobile devices as well as methods to make them serve as better communication support.

References

Boyd-Graber, J., Nikolova, S., Moffatt, K., Kin, K., Lee, J., Mackey, L., Tremaine, M. and Klawe, M. 2006. Participatory design with proxies: developing a desktop-PDA system to support people with aphasia. In *Proceedings CHI '06*, 151–160.

Davies, R., Marcella, S., McGrenere, J. and Purves, B. 2004. The ethnographically informed participatory design of a PDA application to support communication. In *Proceedings of ASSETS '04*, 153–160.

Kauhanen, M.L., Korpelainen, J. T., Hiltunen, P., Määttä, R., Mononen, H., Brusin, E., et al. 2000. Aphasia, depression, and non-verbal cognitive impairment in ischaemic stroke. *Cerebrovascular Diseases*, 10(6): 455–461.

Miller, G. A. 1990. Nouns in WordNet: A lexical inheritance system. *International Journal of Lexicography*, 3(4): 245–264.

Moffatt, K., McGrenere, J., Purves, B., and Klawe, M. 2004. The participatory design of a sound and image enhanced daily planner for people with aphasia. In *Proceedings of CHI '04*, 407–414.

van de Sandt-Koenderman, M., Wiegers, J., and Hardy, P. 2005. A computerized communication aid for people with aphasia. *Disability and Rehabilitation*, 27: 529–533.

A Distributed Database for Mobile NLP Applications*

Petr Homola
Institute of Formal and Applied Linguistics
Charles University
Malostranské náměstí 25
CZ-118 00, Prague, Czech Republic
homola@ufal.mff.cuni.cz

Abstract

The paper presents an experimental machine translation system for mobile devices and its main component — a distributed database which is used in the module of lexical transfer. The database contains data shared among multiple devices and provides their automatic synchronization.

1 Introduction

In Europe, machine translation (MT) is very important due to the amount of languages spoken there. In the European Union, for example, there are more then 20 official languages. Some of them have very few native speakers and it is quite problematic for institutions and companies to find enough translators for comparatively rare language pairs, such as Danish-Maltese. We have developed an experimental MT system for Central and East European languages which is in detail presented in (Homola and Kuboň, 2004); at the moment, we have resources for German, Polish, Czech, Slovak and Russian. As the languages are syntactically and, except of German, lexically related, the system is rule-based. All components of the system are implemented in Objective-C (ObjC) and have been ported to the *iPhone*.

2 Architecture of the MT System

The basic version of the system consists of the following modules:

Morphological analyzer. Since the languages have rich inflection, a word has usually many different endings that express case, number, person etc. It is necessary to assign a lemma and a set of morphological tags to each word form.

Shallow parser. The parser analyzes constituents of the source sentence, but not necessarily whole sentences.

Lexical and structural transfer. The lexical transfer provides a lemma-to-lemma or a term-to-term translation. The structural transfer adapts the syntax of the phrases so that they are grammatical in the target language.

Morphological synthesis of the target language. This final phase generates proper word forms in the target language.

The shallow parser uses the dynamic algorithm described in (Colmerauer, 1969) with feature structures being the main data structure. The hand-written rules are fully declarative and defined in the LFG format (Bresnan, 2001), i.e., they consist of a context-free rule and a set of unificational conditions. The transfer (lexical and structural) is followed by the syntactic and morphological synthesis, i.e., the syntactic structures which represent the source sentences are linearized and proper morphological forms of all words are generated, according to the tag associated with them.

3 Lexical Transfer

The dictionaries are sub-components of the transfer module. Their task is to provide lexical translation of constituents analyzed by the shallow parser. The dictionary contains translation pairs for words and

*The research presented in this paper has been supported by the grant No. 1ET100300517 of the GAAV ČR.

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 26–27,
Columbus, Ohio, USA, June 2008.

phrases. Most items contain an additional morphological or syntactic information such as gender, valence frames etc.

The creation of the dictionaries is a very time-consuming task and they can never cover the complete lexicon of a language. In a production environment, it is inevitable to add new items to the database as new texts are processed. The typical workflow is as follows:

1. During the translation of a document (possibly on a mobile device), unknown words or phrases are found. In the translation, they appear in the source form since the system does not know how to process them. After the processing of the whole document, all found unknown words are added to the database with a remark that the words are new to the system.
2. The new items are transmitted to the computer of a translator whose task is to translate them. Moreover, most items will be assigned a morphological or syntactico-semantical annotation for the structural transfer.
3. The manually updated items are distributed to all instances of application, i.e., to all devices the MT system is installed on, so that they are available for future use by all users of the system.

The capacity of the used mobile device is sufficient to store the lexicon persistently but one could run into problems trying to keep the whole lexicon in memory. For this reason, we use a ternary tree as an index which is kept in memory while full items of the lexicon are loaded from a persistent repository at the moment they are needed.

4 Distributed Database

The database can be used on multiple devices and it is synchronized automatically, i.e., an update of an object is transmitted to all other instances of the database. The synchronization can be deferred if the modifier or the receiver of the update are offline. In such a case, the database is synchronized as soon as the device with the database has access to the internet. Due to the offline synchronization, synchronization conflicts can arise if two or more users update an object simultaneously. If the users have changed different properties of the same object, the changes are merged automatically. Otherwise, the administrator of the database has to resolve the conflict manually.

The distributed database consists of the following components:

Object repository. A local repository of ObjC objects so that the database is accessible even if there is no internet connection.

Transceiver. A communication module that sends/receives updates to/from the relay server. It includes a local persistent cache for updates which is used if there is no internet connection.

Relay server. A server that accepts updates and distributes them to other instances of the database. This component ensures that the database is synchronized even if two or more users are never online at the same time.

It is noteworthy that there is no replica of the database on the server, it only serves as a temporary repository for updated records that cannot be synchronized immediately because a receiving device may be offline at the moment another device has committed an update (this is the expected situation for mobile devices such as PDAs and smartphones).

Currently, the distributed database is being used as a collaboration platform in the Czech Broadcasting Company (Český rozhlas).

5 Conclusions

We have presented an experimental MT system that works on the *iPhone* and described how it uses a distributed object database with automatic synchronization to keep the lexicon of the system up-to-date on all devices it is installed on. We believe that the presented database is an effective way to keep frequently updated data up-to-date on multiple computers and/or mobile devices. The system is developed in Objective-C thus the code base can be used on the *iPhone* and on Macs, and it can be easily ported to systems for which the GNU C Compiler is available.

References

Joan Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers, Oxford.

Alain Colmerauer. 1969. Les systèmes Q ou un formalisme pour analyser et synthétiser des phrases sur ordinateur. Technical report, Mimeo, Montréal.

Petr Homola and Vladislav Kuboň. 2004. A translation model for languages of acceding countries. In *Proceedings of the EAMT Workshop*, Malta.

Author Index