MobiDic – A Mobile Dictation and Notetaking Application

Markku Turunen, Aleksi Melto, Anssi Kainulainen, Jaakko Hakulinen

Speech-based and Pervasive Interaction Group, Tampere Unit for Computer Human Interaction, University of Tampere, Finland

firstname.surname@cs.uta.fi

Abstract

Mobile devices have become ubiquitous and reasonably powerful and well connected. However, their physical size limits possibilities of interaction, especially document creation. Dictation in mobile setting provides one solution, but limited processing power requires that the actual speech recognition is distributed to a server computer. We present MobiDic, a distributed mobile dictation application. Its user interface solutions solve problems that arise from the technical limitation of mobile devices and mobile user context.

Index Terms: mobile applications, dictation, audio editing

1. Introduction

Dictation is one of the few real success stories of speech recognition technology, and speech in general can be very efficient data entry method [1], although there are several challenges for its use [2]. Traditional dictation applications are mostly used by specific professions, e.g., doctors and lawyers, in acoustically safe environments and on computationally adequate hardware. However, people are more and more on the move most of their day, and have immediate access only to less capable hardware, both computationally and interaction-wise. Furthermore, their schedule is often fragmented, so they do not have time for long-lasting dictation sessions to dictate in the traditional sense. People perceive mobile dictation to be very useful, but there are also great demands for its accessibility and usability [3][4]. Thus, mobile solutions allowing incremental dictation and notetaking are necessary to meet the current demands.

However, mobile platforms, and standard mobile phones in particular, are problematic for dictation applications because of lack of processing power. There are only initial prototypes of embedded large-vocabulary speech recognition [5]. Most advanced dictation systems need more hardware capabilities than is available in current and near-future mobile phones. Because of these reasons we have focused on distributed solution, where the dictation and editing can be done with a mobile device but ASR takes place in a server machine. Furthermore, the server stores user audio recordings and recognized texts, and maintains acoustic models and distributes generated document, e.g., via email.

The limited interaction capabilities offered by mobile devices form another fundamental issue. In particular, small screens and keypads make it hard to design efficient document editing interfaces. Most of the desktop applications are assuming standard sized screens and multimodal interaction, in particular when error correction is considered [6]. Furthermore, most of them are targeted for single session, real-time interaction, not asynchronous and fragmented interaction as in mobile dictation applications.

There are some commercial asynchronous mobile dictation solutions available. Typically, the user dictates while on move, and the recognition takes place afterwards. In this case, the process is not interactive, and the user is not able to review, edit and distribute the results in the mobile settings.

We present MobiDic, an asynchronous dictation and notetaking application for mobile phones. In this paper, we describe the MobiDic application, its functionality and architecture, and then discuss the user interface solutions for incremental dictation, audio editing, recognition results editing, and user adaptation.

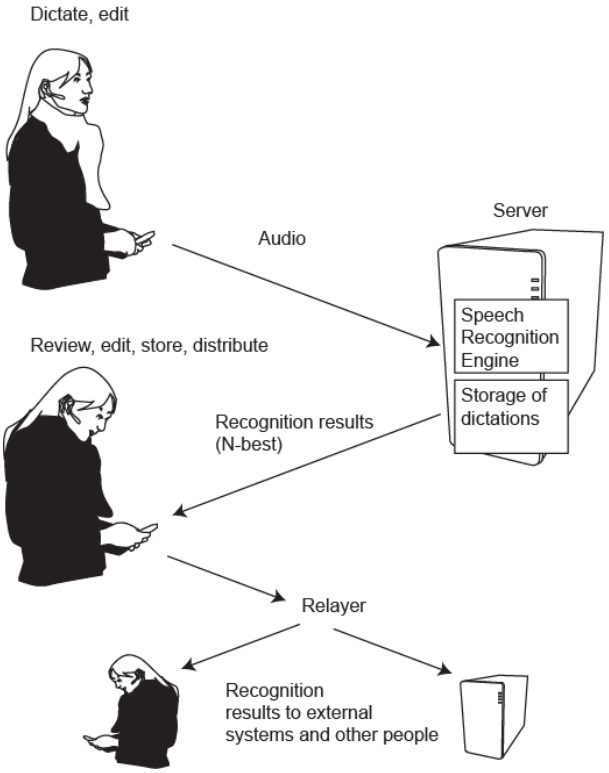

Figure 1. *MobiDic infrastructure.*

2. MobiDic – A Mobile Dictation and Notetaking Application

The overall goal of MobiDic was to design an application that runs on mobile phones and allows specific user groups, such as lawyers considered in the first implementation, to dictate, review, edit, store and distribute dictations and auditory notes and memos efficiently when on the move. In MobiDic, dictation and editing takes place in a mobile device, while automatic speech recognition takes place in a server. Figure 1 illustrates the concept. In the current prototype, Philips Speech SDK 4.2 recognizer (US English models) is used for speech recognition. The memos can be stored both on the

Copyright © 2008 ISCA

server and the mobile device, and they can be distributed to other persons and devices.

Limited analysis of the recorded sound takes place in the mobile phone so that the recordings can be efficiently presented to and edited using novel interaction techniques. The results of the speech recognition can be viewed and edited in the mobile phone, including the possibility of selecting from several alternative recognition results, when available. Next, we present the functionality in more detail.

2.1. MobiDic Functionality

Using MobiDic, audio notes and dictations can be recorded as collections of short recordings. Using short audio segments is suitable for mobile and fragmented use and enables efficient manipulation in the mobile device. Users are directed to make short recordings with user interface solutions. To enable review and editing of the recorded segments, energy-based voice activity detection is used to mark speech and non-speech segments. A graphical editor for manipulating the recordings is included. With it, the user may clean and modify the recordings before they are further processed (recognized in the server, stored or forwarded as such).

Audio recordings can be sent to the server at any point. When requested, recognition results are returned from the server. Server-based recognition is using large vocabulary, domain adapted language models, and speaker specific acoustic models.

A graphical editor is included for review and modification of speech recognition results in a mobile device. Segments ('block') of the document are divided into sentences and words, both of which can contain n-best recognition results. Documents can be stored in mobile devices (e.g., memory card) and the server. Recognition results are aligned with the corresponding audio recording, and augmented with confidence scores and n-best recognition results. The user is able to review and modify both audio and recognition results, with a fluent interface where the user can switch between the recognition results the audio with a single button click.

It should be noted that the combination of audio and recognition results can be used for two purposes. First, the audio files can be source material for speech recognition and the final result will be a text document. Second, speech recognition results can be used to help the user to browse and modify audio recordings. In this case, the result is an audio recording, and even partial recognition is useful. Final documents (auditory, text, or both) can be sent to given recipients (customers, secretaries) as e-mail messages.

3. MobiDic User Interface

MobiDic client application is mobile phone application implemented as a Java 2 Mobile Edition MIDlet. We have used Series 60 compatible mobile phones in development and testing (Nokia N95, E61, 5500). One of the main challenges in MobiDic was to design an efficient interface for recording and reviewing the voice recordings and corresponding recognized text documents. Limited screen space and keyboard require novel interface solutions to efficiently find required locations in the recording for listening and editing.

3.1. Reel User Interface

A novel interface based on the reel metaphor [7] has been implemented (Figure 2) to handle a large number of documents and blocks within documents. Reel menus are two dimensional fish-eye menus that can be navigated with the directional buttons of a mobile phone. Items in a menu are on top of each other and the user can roll the reel to select menu items. The currently selected node is enlarged to provide more information and making it easier to see the information on the small display. The adjacent items (both horizontally and vertically) are visible so that the user always has a context for the current selection, supporting navigation in a large amount of documents. The size and amount of visible menu nodes depend on the phone model, which enables scalability for screens with different size. Directional buttons are used for most of the interaction, e.g., menu navigation, node selection, cursor movement in audio and text views. Softkey menus are used for extra commands and the keypad for additional text input.

Figure 2. *The reel menu metaphor*

The fish-eye reel makes it possible to use text-to-speech together with graphical outputs. The content of each item in a reel can be read out loud by the speech synthesizer when the item is activated. In this way, the application can be used without seeing the screen. This can be useful in mobile settings when reviewing recognition results, for example.

3.2. Document creation

Figure 2 illustrates how document creation, audio settings and the different documents can be found on the main menu with up and down navigation of the reel. Different views or parts of the dictation process (audio recording and editing, text editing) can be found with sideways navigation of a sub-reel. After creating a new document, the user can make one or more recordings within that document. New audio can be recorded at any time, even after some have already been recognized and edited. This allows incremental and open ended dictation, since the user can easily return to the task later on.

In desktop applications, recordings can possibly be tens of minutes long. Here, the user interface encourages the users to limit the recording to much shorter segments, about two sentences per recording. This is because shorter recordings make it easier to edit or add new the audio data in between others later on. Many recording mistakes are actually corrected by simply deleting and re-recording a block. Shorter recordings are also preferable for asynchrony with the recognizer server, since transferring the data of multiple smaller recordings is more suitable for slow connections in fragmented usage sessions. The recording view consists of a simple counter and bar representing the length of the recording so far. The growing progress bar is a sneak approach to guiding users to make shorter recordings. Recordings are done at the sample rate of 8000 Hz and the resolution of 16 bit, so in mono they take 16 kilobytes of storage space per second of audio data.

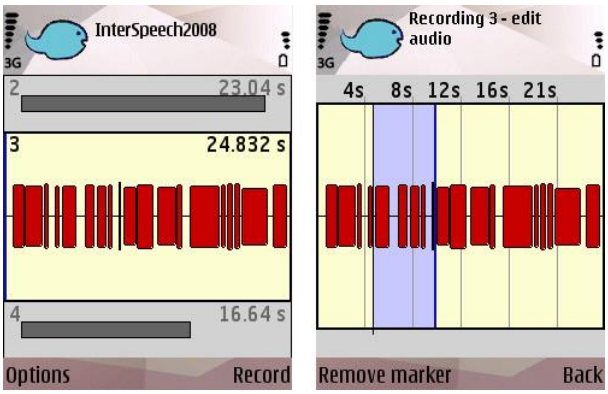

Figure 3: Left: *Audio blocks of a recording. Right: Editing an audio block with markers.*

Figure 3 portrays how multiple audio blocks are presented in the document reel on top of each other as bars representing total recording times. They can be navigated just like the main menu. The lengths of the unselected bars are relative to each other, but are scaled to fit to the screen. The current chosen block shows a more detailed representation of the audio data, given as columns of various sizes. The columns represent short utterances, or a few words, as detected by Voice Activity Detection (VAD) algorithm. The visualization requires less knowledge of acoustics than the common time-amplitude waveform plot. The VAD algorithm, based on measuring mean sound energy levels, is lightweight and near-instantaneous (a five second recording taking less than a second to appear after recording is done, and a 30 second one taking one to two seconds). It is possible to play aloud the recordings with a cursor helping to create a connection between visual and auditory representations.

3.3. Audio editor

Since audio recordings often contain hesitations, misspoken words, speech targeted to someone else, noise, and lengthy sections of silence, MobiDic includes an audio editor for editing the recorded blocks. Cleaning raw recordings improves recognition results, if recognized text is the final form, or makes audio notes ready to be used as such. The editor view, as seen on Figure 3, resembles the audio block view, but shows only the selected block. A cursor can be moved with playback or by jumping between VAD start and end points. The view includes tick marks to help estimate selection. The user can delete a section within the block by setting start and end markers. In addition, a new section can be recorded for the selected point. Removed sections are actually not deleted from audio data. Instead, a markup language (see Section 4) is used both at the server and the mobile device to contain changes and all data is stored to allow lossless modifications.

3.4. Speech Recognition Server

After blocks have been recorded and optionally edited, they can be sent to be recognized one by one or all at the same time. To enable asynchronous and even concurrent processing between the server and the client, recognition results can be fetched as they come available. The server recognizes audio files as they are sent from the mobile client. It removes the marked segments from the audio files just before the audio is fed to the recognizer.

Philips Speech SDK 4.2 has been used to develop the system, but any similar recognizer can be used instead. User specific acoustic model and domain adapted language model are used. Recognition results, including n-best lists and confidence scores are served back to the client when it requests them. Speech recognition works faster than real time so in practice all the delays related to the distributed recognition come from data transmission. The server enables concurrent uploading of audio, speech recognition, adaptation, and downloading of recognition results.

3.5. Recognition Result Editor

As seen in Figure 6, the recognized text retains the division into documents and blocks. Text blocks within documents can be viewed in a reel, just like the auditory blocks. The blocks of recognition results are divided into sentences, which in turn are divided into words. Both sentences and words are given confidence scores. For the user, this is displayed using a color coding based on confidence scores, which employs a simple gamut, with higher confidence resembling regular text, and worse confidence standing out more clearly.

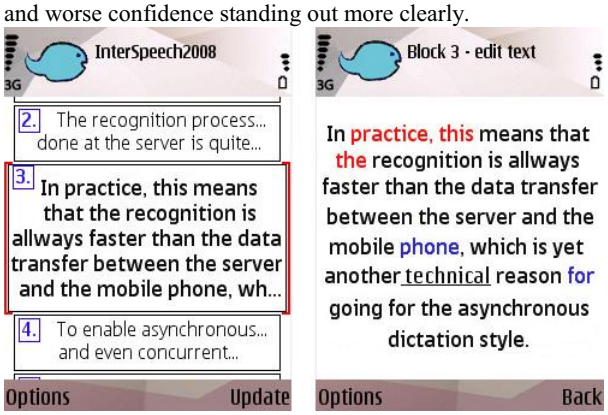

Figure 4: Left: Several text blocks in a document, Right: editing a text block.

Text can be navigated one word at a time and row by row. Selecting a word shows an n-best list of words as a reel menu, as seen in Figure 5, from which the correct word can be selected, or a new one added replacing the selection using normal text input methods of the mobile device. Selected words can also be deleted and new ones inserted between existing words.

A full document preview is also available to quickly see the final results. It functions as an email application, allowing the user to write comments and send the document as an e-mail message via server.

3.6. Training the Recognizer

The MobiDic application contains adapted language models to better suit the language used by the target user groups. In addition, current dictation recognizers require speaker specific acoustic training. Without any training, our initial tests resulted WER over 60%, one training set of about 10 minutes decreasing this to less than 20%, and yet another set resulted error rates of 10%. These numbers contain some errors caused by out-of-vocabulary errors remedied by adapting the language model. Users can train an acoustic model for themselves by completing training lessons suitable for mobile usage, as illustrated in Figure 5. Users are able to download the training texts, then record the exact dictations, and finally send them to server for acoustic adaptation.

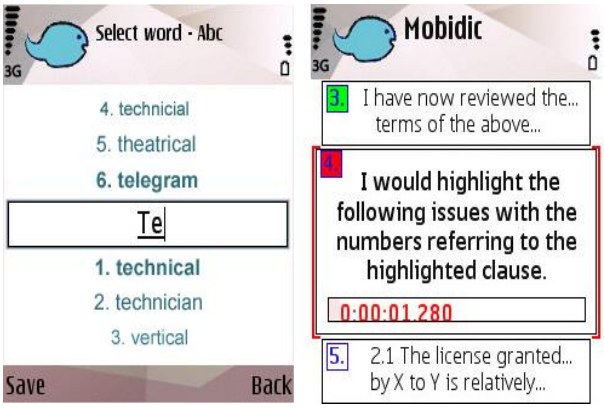

Figure 5: Left: Editing word in a reel-based n-best list. Right: Recognizer training interface.

Corrected recognition results can be sent back to the server to be used to adapt the acoustic model per user. At the moment, the adapting feature is not allowed if the text is modified so it contains other words, e.g., added ones, than those spoken. This is to make sure mismatching spoken and written content are not used for adaptation.

4. A Lightweight Markup Language

In order to implement the MobiDic application we defined a markup language suitable for mobile dictations and auditory notetaking. After studying existing markup languages for annotation of spoken and multimodal data, such as Annotations Graphs [8], we decided to create a lightweight alternative for this purpose, as presented in the following document type declaration:

```
<!ELEMENT document (block+)>
<!ELEMENT block (remove?, sentence*, vad?)>
<!ELEMENT remove (voice+)>
<!ELEMENT sentence (word+)>
<!ELEMENT word (#PCDATA)>
<!ELEMENT vad (voice*)>
<!ELEMENT voice EMPTY>
<!ATTLIST document
        id ID #REQUIRED
        name CDATA #REQUIRED>
<!ATTLIST block
        id ID #REQUIRED
        position CDATA #IMPLIED
        audio CDATA #REQUIRED
        start CDATA "0"
        end CDATA #IMPLIED
        status (ORIG|MODIFIED|NO-TRAINING) "ORIG">
<!ATTLIST sentence
        id ID #IMPLIED
        position CDATA #IMPLIED
        start CDATA #IMPLIED
        end CDATA #IMPLIED
        score CDATA #IMPLIED
        status (ORIG|MODIFIED) "ORIG">
<!ATTLIST word
        id ID #IMPLIED
        position CDATA #IMPLIED
        start CDATA #IMPLIED
        end CDATA #IMPLIED
        score CDATA #IMPLIED
        status (ORIG|DELETED|INSERTED|REPLACED|MOVED)
"ORIG"
        userSelected (true|false) "false">
<!ATTLIST vad
        id ID #IMPLIED>
<!ATTLIST voice
        start CDATA #REQUIRED
        end CDATA #REQUIRED
        value CDATA #IMPLIED>
```

The purpose of the lightweight markup language is to allow efficient implementation in the mobile device, but still make it flexible enough solution to contain n-best recognition results both the word and sentence levels. This is necessary, since different recognizers give n-best results in different segments. Furthermore, it is possible to mark all editions (deletions, insertions and modifications), and the speech/silence segmentation based on voice activity detection.

5. Conclusions and Future Work

We have presented MobiDic, an application, which enables mobile dictation and notetaking. In MobiDic, recording, recognition and editing of recognition results are distributed between a mobile client and a server. MobiDic handles the challenges of mobile dictation with an interface that guides users to construct documents from short recordings which can be efficiently browsed, edited and stored in the mobile device. The application includes two tightly integrated editors, the audio editor and the recognition results editor, to efficiently edit and review audio recordings and corresponding speech recognition results. We also presented a lightweight markup-language for such applications.

In the future, the MobiDic application will be tested in real usage environments in UK and Finland. Pilot customer tests consist of users who move a lot during their workday. The pilots will last for a period of between two to four weeks. Both objective and subjective data will be collected from the pilot use of the system.

6. Acknowledgements

The development of the MobiDic application was made possible by Mobiter Dicta Oy, the future developer of the application. We thank Ville Antila and Tomi Heimonen for their contribution in the implementation.

7. References

[1] Moore, R. K., Modelling data entry rates for ASR and alternative input methods, Proc. INTERSPEECH 2004 ICSLP, 2004.

[2] Moore R. K. Research Challenges in the Automation of Spoken Language Interaction. In Proceedings of Workshop on Applied Spoken Language Interaction in Distributed Environments (ASIDE 2005) 2005.

[3] Basapur, S, Xu, S., Ahlenius, M., Lee, Y.S., User Expectations from Dictation on Mobile Devices, Human-Computer Interaction. Interaction Platforms and Techniques, LNCS, 4551/2007.

[4] Cox, A., Walton, A. Evaluating the viability of speech recognition for mobile text entry. In: Proceedings of HCI 2004: Design for Life, pp. 25–28 (2004)

[5] Karpov, E. , Kiss, I. , Leppänen, J. , Olsen, J. , Oria, D. , Sivadas, S. and Tian, J., "Short Message Dictation on Symbian Series 60 Mobile Phones". In Proceedings of MobileHCI 2006 Workshop on Speech in Mobile and Pervasive Environments, 2006.

[6] Suhm, B., Myers, B., Waibel, A. Multimodal error correction for speech user interfaces. ACM Transactions on Computer-Human Interaction (TOCHI), March 2001, 8(1), 2001.

[7] Turunen, M., Hakulinen, J., Kainulainen, A., Melto, A., and Hurtig, T. Design of a Rich Multimodal Interface for Mobile Spoken Route Guidance. In Proceedings of Interspeech 2007 - Eurospeech: 2193-2196, 2007.

[8] Bird, S., Liberman, M. A Formal Framework for Linguistic Annotation. Speech Communication 33(1-2): 23-60, 2001.