A Speech-Centric Perspective for Human-Computer Interface

L. Deng, A. Acero, Y. Wang, K. Wang, H. Hon, J. Droppo, M. Mahajan, and X.D.Huang
Microsoft Research,
One Microsoft Way, Redmond, WA 98052, USA.

Abstract— **Speech technology has been playing a central role in enhancing human-machine interactions, especially for small devices for which GUI has obvious limitations. The speech-centric perspective for human-computer interface advanced in this paper derives from the view that speech is the only natural and expressive modality to enable people to access information from and to interact with any device. In this paper, we describe the work conducted at Microsoft Research, in the project codenamed *Dr.Who*, aimed at the development of enabling technologies for speech-centric multimodal human-computer interaction. In particular, we present MiPad as the first *Dr. Who's* application that addresses specifically the mobile user interaction scenario. MiPad is a wireless mobile PDA prototype that enables users to accomplish many common tasks using a multimodal spoken language interface and wireless-data technologies. It fully integrates continuous speech recognition and spoken language understanding, and provides a novel solution to the current prevailing problem of pecking with tiny styluses or typing on minuscule keyboards in today's PDAs or smart phones.**

Keywords—human-computer interaction; speech-centric multimodal interface; robust speech recognition; spoken language understanding; MiPad.

I. INTRODUCTION

Speech technology has been playing a key role in enabling and enhancing human-machine communications. In combination with multimedia processing technologies, speech processing will in the future also contribute, in a significant way, to facilitating human-human interactions. In applications such as distributed meetings, audio-visual browsing, and multimedia annotations, automatic processing of natural, spontaneous speech will collaborate with automatic audio-visual object tracking and other multimedia processing techniques to complete full end-to-end systems. In addition to the multimedia applications, the most important role that speech can play is in a full range of the devices that demand efficient human inputs. Since speech is the only natural and expressive modality for information access from and interaction with any device, we highlight the speech-centric view of human-machine interface.

Graphic user interface (GUI), based primarily on the exploitation of visual information, has significantly improved computer-human interactions by using intuitive real-world metaphors. However, it is still far from achieving the ultimate goal of allowing users to interact with the computer without training. In particular, GUI relies heavily on a sizeable screen, keyboard, and pointing device, which are not always available. In addition, with more and more computers designed for mobile usages and hence subject to the physical size and hands-busy or eyes-busy constraints, the traditional GUI faces an even greater challenge. Multimodal interface enabled by spoken language is widely believed to be capable of dramatically enhancing the usability of computers because GUI and speech have complementary strengths. While spoken language has the potential to provide a natural interaction model, the ambiguity of spoken language and the memory burden of using speech as output modality on the user have so far prevented it from becoming the choice of mainstream interface. Multimodal Intelligent Personal Assistant Device, or MiPad [5], is one of our attempts in overcoming such difficulties by developing enabling technologies for speech-centric multimodal interface. MiPad is a prototype of wireless Personal Digital Assistant (PDA) that enables users to accomplish many common tasks using a multimodal spoken language interface (speech + pen + display).

In this paper, we will describe the work carried out at Microsoft Research on several major component technologies underlying MiPad. These include: distributed and robust speech processing; acoustic and language modeling for accurate speech recognition; robust parser for processing the speech recognizer's output; schema-based knowledge representation for MiPad's PIM (personal information management) task; *Tap & Talk* multimodal interaction and user interface design; back channel communication and MiPad's error repair strategy.

II. DISTRIBUTED AND ROBUST SPEECH PROCESSING

Robustness to acoustic environment (i.e. immunity to noise and channel distortion) is one most important aspect of the MiPad design considerations. For MiPad to be acceptable to the general public, it is desirable to remove the need for a close-talking microphone in capturing speech. Although close-talking microphones pick up relatively little background noise and allow speech recognizers to achieve high accuracy for the MiPad-domain tasks, it is found that users much prefer built-in microphones even if there is minor accuracy degradation. With the convenience of using built-in microphones, noise robustness becomes a key challenge to maintaining desirable speech recognition and understanding performance. Our recent work on speech processing aspects of MiPad has focused on this noise-robustness challenge in the framework of distributed speech recognition (DSR) that MiPad design has adopted.

0-7803-7713-3/02/$17.00 © 2002 IEEE

Authorized licensed use limited to: MICROSOFT. Downloaded on December 24, 2008 at 03:08 from IEEE Xplore. Restrictions apply.

There has recently been a great deal of interest in standardizing DSR applications for a plain phone, PDA, or a smart phone where speech recognition is carried out at a remote server. To overcome bandwidth and infrastructure cost limitations, one possibility is to use a standard codec on the device to transmit the speech to the server where it is subsequently decompressed and recognized. However, since speech recognizers such as the one in MiPad only need some features of the speech signal (e.g., Mel-cepstrum), bandwidth can be further saved by transmitting only those features. Our recent work on noise robustness directly relevant to the MiPad front end has been concentrated on the Aurora2 and 3 tasks [4], an effort to standardize a DSR front-end that addresses the issues surrounding robustness to noise and channel distortions at a low bit rate .

In DSR applications, it is easier to update software on the server because one cannot assume that the client is always running the latest version of the algorithm. With this consideration in mind, while designing noise-robust algorithms for MiPad, we strive to make the algorithms front-end agnostic. That is, the algorithms should make no assumptions on the structure and processing of the front end and merely try to undo whatever acoustic corruption that has been shown during training. This consideration also favors noise-robust approaches in the feature rather than in the model domain.

We have developed several high-performance speech feature enhancement algorithms on the Aurora2 and 3 tasks and on other Microsoft internal tasks with much larger vocabularies. One most effective algorithm is called SPLICE, short for Stereo-based Piecewise Linear Compensation for Environments [1][2][3]. In a DSR system, the SPLICE may be applied either within the front end on the client device, or on the server, or on both with collaboration. Certainly a server side implementation has some advantages as computational complexity and memory requirements become less of an issue and continuing improvements can be made to benefit even devices already deployed in the field. Another useful property of SPLICE in the serve implementation is that new noise conditions can be added as they are identified by the server. This can make SPLICE quickly adapt to any new acoustic environment with minimum additional resource.

III. ACOUSTIC AND LANGUAGE MODELS FOR CONTINUOUS SPEECH RECOGNITION

MiPad is designed to be a *personal* device. As a result, the recognition uses speaker-adaptive acoustic models (HMMs) and a user-adapted lexicon to improve recognition accuracy. The HMMs and the continuous speech decoding engine are both derived from an improved version of the Microsoft's Whisper speech recognition system and of the HTK, which combines the best features of these earlier two separate systems. Both MLLR and MAP techniques are used to adapt the speaker-independent acoustic model for each individual speaker. The acoustic model contains about 6000 senones, each with 20-component mixture Gaussian densities. The context-sensitive language model is used for relevant semantic objects driven by the user's pen tapping action. As speech recognition accuracy remains as a major challenge for MiPad usability, most of our recent work on MiPad's acoustic modeling has focused on noise robustness we just outlined. The work on language modeling for improving speech recognition accuracy has focused on language model portability which we elaborate below.

The speech recognition engine in MiPad uses the unified language model that takes advantage of both rule-based and data-driven approaches [7]. Consider two training sentences:

"Meeting at three with Zhou Li". vs.

"Meeting at four PM with Derek".

Within a pure n-gram framework, we will need to estimate

P(Zhou|three with) and *P(Derek|PM with)*

individually. This makes it very difficult to capture the obviously needed long-span semantic information in the training data. To overcome this difficulty, the unified model uses a set of CFGs that captures some of the common named entities. For the example listed here, we may have CFG's for <NAME> and <TIME> respectively, which can be derived from the factoid grammars of smaller sizes. The training sentences now look like:

"Meeting <at three:TIME> with <Zhou Li:NAME>", and

"Meeting <at four PM:TIME> with <Derek: NAME>".

With parsed training data, we can now estimate the n-gram probabilities as usual. For example, the replacement of

P(Zhou|three with) ← P(<NAME>|<TIME> with)

makes such "n-gram" representation more meaningful and more accurate.

Inside each CFG, however, we can still derive

P("Zhou Li"|<NAME>) and P("four PM"|<TIME>)

from the existing n-gram (n-gram probability inheritance) so that they are appropriately normalized. This unified approach can be regarded as a generalized n-gram in which the vocabulary consists of words and structured classes.

Most decoders can only support either CFGs or word n-grams. These two ways of representing sentence probabilities were mutually exclusive. We have modified the decoder so that we can embed CFGs in the n-gram search framework to take advantage of the unified language model. An evaluation of the use of the unified language model is shown in Table 1. The speech recognition error rate with the use of the unified language model is demonstrated to be significantly lower than that with the use of the domain-independent trigram. That is, incorporating the CFG into the language model drastically improves cross-domain portability. The test data shown in Table 1 are based on MiPad's PIM *conversational speech*. The domain-independent trigram language model is based on Microsoft Dictation trigram models used in Microsoft Speech SDK 4.0. In Table 1, we also observe that using the unified language model directly in the decoding stage produces about 10% fewer recognition errors than doing N-best re-scoring using the identical language model. This demonstrates the

Authorized licensed use limited to: MICROSOFT. Downloaded on December 24, 2008 at 03:08 from IEEE Xplore. Restrictions apply.

importance of using the unified model in the early stage of speech decoding, at least for the MiPad's PIM task.

TABLE I: CROSS-DOMAIN (DICTATION TO PIM) RECOGNITION PERFORMANCE WITH THE UNIFIED LANGUAGE MODEL

Systems	Perplexity	Word Error
Domain-indept Trigram	593	35.6%
Decoder with Unified LM	141	22.5%
N-best re-scoring with unified LM	-	24.2%

IV. SPOKEN LANGUAGE UNDERSTANDING AND DIALOGUE

The spoken language understanding (SLU) engine used in our speech-centric multimodal human-computer interaction research, MiPad research in particular, is based on a robust chart parser [6] and a plan-based dialog manager [8]. Each semantic object defined and used for SLU is either associated with a real-world entity or an action that the application takes on a real-entity. Each semantic object has slots that are linked to their corresponding CFG. In contrast to the sophisticated prompting response in voice-only conversational interface, the response is a direct graphic rendering of the semantic object on MiPad's display. After a semantic object is updated, the dialog manager fulfills the plan by executing the application logic and the error repair strategy.

A. *Semantic schema*

MiPad adopts a semantic based robust understanding technology for spoken language understanding. At the center of the technology is *semantic schema* defined in the Semantic Description Language (SDL). The semantic schema is a domain model; it defines the entity relations of a specific domain. The semantic schema is used for many different purposes. It serves as the specification for a language-enabled application: once a semantic schema is defined, grammar and application logic development can proceed simultaneously according to the semantic schema. It also plays a critical role in dialogue management. Further, semantic schema is language and expression independent in the sense that it does not specify the linguistic expressions used to express the concepts. Because of this, it is used not only for language-enabled applications, but also for integrating inputs from multi-modalities, such as mouse click events.

A semantic schema consists of a list of definitions for semantic classes. A semantic class corresponds to a concept in the application domain. Slots of a semantic class are specified with either a type or a semantic class. The former constrains that the slot must be filled with a semantic object (an instantiation of a semantic class) that has the specified type, and the latter requires that the slot be filled with an instantiation of that specific semantic class. In case two slots are specified with the same type or semantic class, additional names are used to differentiate them.

In a human-machine conversation, the computer system responds to the semantics of a user's utterance (word sequence) with an appropriate action. It does so with the help of the discourse structure, which accumulates over all the relevant semantic information from the beginning of the conversation up to the current utterance. Both the utterance semantics and the discourse information are represented in an XML structure that maps words in the utterances to the semantic classes (including all their slots).

B. *Robust parser and SLU in MiPad*

Since the *Tap & Talk* interface in MiPad explicitly provides dialog state (tapped field) information already, dialogue management plays relatively minor role, compared with the SLU, in the overall MiPad functionality. The major SLU component in MiPad is a robust chart parser, which accepts the output of the continuous speech recognizer using field-specific language models and employs field-specific grammars. In the typical MiPad usage scenario, users use the built-in MiPad microphone that is very sensitive to environment noise. With the iPaq device from Compaq as one of our prototypes, the word recognition error rate has increased by a factor of two compared with a close-talking microphone in the normal office environment. This highlights the need not only for noise-robust speech recognition but also for robust SLU.

The MiPad SLU is built on domain-specific semantic grammars. Normally, semantic grammars are CFGs with non-terminals representing semantic concepts instead of syntactic categories. Our grammars introduce a specific type of non-terminals called semantic classes to describe the schema of an application. The semantic classes define the conceptual structures of the application that are independent of linguistic structures. The linguistic structures are modeled with CFGs. In doing so, it makes the linguistic realization of semantic concepts transparent to an application; therefore the application logic can be implemented according to the semantic class structure, in parallel with the development of linguistic CFGs. We in the past few years have developed a robust spoken language parser that analyzes input sentences according to the linguistic grammar and maps the linguistic structure to the semantic conceptual structure. Recently, we have made substantial modifications to the parser to take full advantage of the form factor of MiPad and to better support the semantics-based analysis.

The robust parsing algorithm used in MiPad is an extension of the bottom-up chart-parsing algorithm. The robustness to ungrammaticality and noise can be attributed to its ability of skipping minimum unparsable segments in the input. The algorithm uses dotted rules, which are standard BNF/CFG rules plus a dot in front of a right-hand-side symbol. The dot separates the symbols that already have matched with the input words from the symbols that are yet to be matched. Each constituent constructed in the parsing process is associated with a dotted rule. If the dot appears at the end of a rule like in $A \rightarrow \alpha\bullet$, , we call it a complete parse

Authorized licensed use limited to: MICROSOFT. Downloaded on December 24, 2008 at 03:08 from IEEE Xplore. Restrictions apply.

with symbol **A**. If the dot appears in the middle of a rule like in **A→B•CD**, we call it a partial parse (or hypothesis) for A that is expecting a complete parse with root symbol **C**.

The algorithm maintains two major data structures --- A chart holds hypotheses that are expecting a complete constituent parse to finish the application of the CFG rules associated with those hypotheses; an agenda holds the complete constituent parses that are yet to be used to expand the hypotheses in the chart. Initially the agenda is empty. When the agenda is empty, the parser takes a word (from left to right) from the input and puts it into the agenda. It then takes a constituent $\mathbf{A}[i,j]$ from the agenda, where A is the root symbol of the constituent and $[i,j]$ specifies the span of the constituent. The order by which the constituents are taken out of the agenda was discussed in [6]. The parser then activates applicable rules and extends appropriate partial parses in the chart. A rule is applicable with respect to a symbol **A** if either A starts the rule or all symbols before **A** are marked optional. The activation of an applicable rule may result in multiple constituents that have the same root symbol (the left-hand-side of the rule) but different dot positions, reflecting the skip of different number of optional rule symbols after **A**. If the resulting constituent is a complete parse, namely with the dot positioned at the end of the rule, the complete constituent is added into the agenda. Otherwise partial constituents are added into the chart; To extend the partial parses with the complete parse $\mathbf{A}[i,j]$, the parser exams the chart for incomplete constituent with dotted rule $\mathbf{B}[l,k] \rightarrow \alpha \bullet \mathbf{A}\beta$ for $k < i$, and constructs new constituents $\mathbf{B}[l,j] \rightarrow \alpha\mathbf{A}\beta$ with various dot positions in β, as long as all the symbols between **A** and the new dot position are optional. The complete constituent $\mathbf{B}[l,j] \rightarrow \alpha\mathbf{A}\beta\bullet$ is added into the agenda. Other constituents are put into the chart. The parser continues the above procedure until the agenda is empty and there are no more words in the input sentence. By then it outputs top complete constituents according to some heuristic scores.

V. MIPAD USER INTERFACE DESIGN

MiPad takes advantage of the graphical display in the UI design. The graphical display simplifies dramatically the dialog management. For instance, MiPad is able to considerably streamline the confirmation and error repair strategy as all the inferred user intentions are confirmed *implicitly* on the screen. Whenever an error occurs, the user can correct it in different modalities, either by soft keyboard or speech. The user is not obligated to correct errors immediately after they occur. The display also allows MiPad to confirm and ask the user many questions in a single turn. Perhaps the most interesting usage of the display, however, is the *Tap & Talk* interface which we discuss now.

A. Tap & Talk interface

Because of MiPad's small form-factor, the present pen-based methods for getting text into a PDA (Graffiti, Jot, soft keyboard) are potential barriers to broad market acceptance. Speech is generally not as precise as mouse or pen to perform position-related operations. Speech interaction can also be adversely affected by the unexpected ambient noise, despite the use of denoising algorithms in MiPad. Moreover, speech interaction could be ambiguous without appropriate context information. Despite these disadvantages, speech communication is not only natural but also provides a powerful complementary modality to enhance the pen-based interface if the strengths of using speech can be appropriately leveraged and the technology limitations be overcome. The advantage of pen is typically the weakness of speech and vice versa.

Tap & Talk is a key feature of the MiPad's user interface design. The user can give commands by tapping the *Tap & Talk* field and talking to it. *Tap & Talk* avoids speech detection problem that are critical to the noisy environment deployment for MiPad. The appointment form shown on MiPad's display is similar to the underlying semantic objects. By tapping to the *attendees* field in the calendar card, for example, the semantic information related to potential attendees is used to constrain both CSR and SLU, leading to a significantly reduced error rate and dramatically improved throughput. This is because the perplexity is much smaller for each slot-dependent language and semantic model. In addition, *Tap & Talk* functions as a user-initiative dialog-state specification. The dialog focus that leads to the language model is entirely determined by the field tapped by the user. As a result, even though a user can navigate freely using the stylus in a pure GUI mode, there is no need for MiPad to include any special mechanism to handle spoken dialog focus and digression.

B. Back Channel Communications

MiPad handles back-channel communications on the device. As a user speaks, it displays a graphical meter reflecting the volume of the recording. When the utterance is beyond the normal dynamic range, red bars are shown to instruct the user to tone down. As the host computer processes the user's utterance, a running status bar is shown. The user can click a cancel button next to the status bar to stop the processing at the host computer. If the status bar vanishes without changing the display, it indicates the utterance has been rejected either by the recognizer or by the understanding system. MiPad's error repair strategy is entirely user initiative: the user can decide to try again or do something else.

C. User study results

Our ultimate goal is to make MiPad produce real value to users. It is necessary to have a rigorous evaluation to measure the usability of the prototype. Our major concerns are: "*Is the task completion time much better*?" and "*Is it easier to get the job done*?"

For our user studies, we set out to assess the performance of the current version of MiPad (with PIM features only) in terms of task-completion time, text throughput, and user satisfaction. In this evaluation, computer-savvy participants who had little experience with PDAs or speech recognition software used the partially implemented MiPad prototype. The tasks we evaluated include creating a new appointment and creating a new email. Each participant completed half the tasks using the *Tap & Talk* interface and half the tasks using the regular pen-only iPaq interface. The ordering of *Tap & Talk* and pen-only tasks is statistically balanced.

Authorized licensed use limited to: MICROSOFT. Downloaded on December 24, 2008 at 03:08 from IEEE Xplore. Restrictions apply.

1) Is the task completion time much better?

Twenty subjects were included in the experiment to evaluate the tasks of creating a new email, and creating a new appointment. Task order was randomized. We alternated tasks for different user groups using either pen-only or *Tap & Talk* interfaces. The text throughput is calculated during e-mail paragraph transcription tasks. On average it took the participants 50 seconds to create a new appointment with the *Tap & Talk* interface and 70 seconds with the pen-only interface. This result is statistically significant with $t(15) = 3.29$, $p < .001$. The saving of time is about 30%. For transcribing an email it took 2 minutes and 10 seconds with *Tap & Talk* and 4 minutes and 21 seconds with pen-only. This difference is also statistically significant, $t(15) = 8.17$, $p < .001$. The saving of time is about 50%. Error correction for the *Tap & Talk* interface remains as one of the most unsatisfactory features. In our user studies, calendar access time using the *Tap & Talk* methods is about the same as pen-only methods, which suggests that pen-based interaction is suitable for simple tasks.

2) Is it easier to get the job done?

Fifteen out of the 16 participants in the evaluation stated that they preferred using the *Tap & Talk* interface for creating new appointments and all 16 said they preferred it for writing longer emails. The preference data is consistent with the task completion times. Error correction for the *Tap & Talk* interface remains as one of the most unsatisfactory features. On a seven point Likert scale, with 1 being "disagree" and 7 being "agree", participants responded with a 4.75 that it was easy to recover from mistakes.

VI. SUMMARY AND CONCLUSION

The speech-centric perspective for human-machine interface is based on the recognition that speech is a necessary modality to enable a pervasive and consistent user interaction with computers across a full range of devices --- large or small, fixed or mobile, and that speech has the potential to provide a natural user interaction model. However, the ambiguity of spoken language, the memory burden of using speech as output modality on the user, and the limitations of current speech technology have prevented speech from becoming the choice of mainstream interface. Multimodality is capable of dramatically enhancing the usability of speech interface because GUI and speech have complementary strengths as we have shown in this paper. Multimodal access will enable users to interact with an application in a variety of ways --- including input with speech, keyboard, mouse and/or pen, and output with graphical display, plain text, motion video, audio, and/or synthesized speech.

Dr. Who is Microsoft's attempt to develop a speech-centric multimodal user interface framework and its enabling technologies. MiPad as we have focused on in this paper is the first *Dr. Who's* application that addresses specifically the mobile interaction scenario and it aims at the development of a consistent human-computer interaction model and component technologies for multimodal applications. Our current applications comprise mainly the PIM functions. Despite its current incomplete implementation, we have observed that speech and pen have the potential to significantly improve user experience in our user study. Thanks to the multimodal interaction, MiPad also offers a far more compelling user experience than standard voice-only telephony interaction.

The success of MiPad depends on spoken language technology and an always-on wireless connection. With upcoming 3G wireless deployments in sight, the critical challenge for MiPad remains the accuracy and efficiency of our spoken language systems. This is because most likely MiPad will be used in the noisy environment with no availability of a close-talking microphone, which demands a high degree of speech recognition and parsing accuracy. The efficiency need arises because the server will be supporting a large number of MiPad clients. To meet this challenge, much of our recent work has focused on: 1) noise-robustness and transmission efficiency aspects of the MiPad system in the distributed speech processing environment, and 2) SLU with specific attention paid also to robustness as well as to automatic and high-quality application grammar development.

The prototype of MiPad as the first *Dr. Who* application discussed in this paper has recently been successfully transferred from the research lab to the Microsoft .NET speech product division as a client browser component in the grand Kokanee architecture. This new architecture is aimed at speech-enabling the web applications based on the Speech-Application-Language-Tag standard for multimodal (speech and GUI) interactions between end users and either mobile or fixed devices. Future research work will be focused on the next version of *Dr. Who* and its new applications aimed to provide a greater degree of intelligence and automation to larger domains and tasks than the limited PIM task of the MiPad developed so far.

REFERENCES

[1] L. Deng, A. Acero, M. Plumpe, and X.D. Huang. "Large-vocabulary speech recognition under adverse acoustic environments," Proc-ICSLP2000, Beijing, China, October 2000, Vol. 3, p. 806-809

[2] L. Deng, A. Acero, L. Jiang, J. Droppo, and XD Huang. "High-performance robust speech recognition using stereo training data," Proc. ICASSP-2000, Vol. I, Salt Lake City, Utah, April 2001, pp. 301-304.

[3] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora2 and 3 databases," *Proc. ICSLP-2002, in press.*

[4] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium,"* Paris, France, September 18-20, 2000.

[5] X. D. Huang et al. "MiPad: A multimodal interaction prototype," *Proc. ICASSP-2001*, Vol. I, Salt Lake City, Utah, April 2001, p. 9-12.

[6] Y. Wang, "A robust parser for spoken language understanding," *Proc. Eurospeech-1999*, Budapest, Hungary, 1999.

[7] Y. Wang, M. Mahajan, X. Huang, "A unified context-free grammar and N-gram model for spoken language processing", *Proc. ICASSP-2000*, Istanbul, Turkey, 2000.

[8] K. Wang, "Implementation of a multimodal dialog system using extended markup language," *Proc. ICSLP-2000*, Beijing, China, 2000.

Authorized licensed use limited to: MICROSOFT. Downloaded on December 24, 2008 at 03:08 from IEEE Xplore. Restrictions apply.