SEMANTICS SYNCHRONOUS UNDERSTANDING FOR ROBUST SPOKEN LANGUAGE APPLICATIONS

Kuansan Wang

Speech Technology Group, Microsoft Research, Redmond, WA 98052, USA

ABSTRACT

In this paper, we describe our recent effort in combining the speech recognition and understanding into a single pass decoding process. The goal is to utilize the semantic structure not only to better handle disfluencies and improve the overall understanding accuracy, but also to shorten the response time and achieve higher interactivity. Three related techniques are instrumental in our approach. First, we employ the unified language model (ULM) to incorporate semantic schema into the recognition language model, and extend the search process from word synchronous to semantic object synchronous (SOS) decoding. Finally, we utilize sequential detection to defer, reject, or accept semantic hypotheses and execute consequent dialog actions while the user's utterance is ongoing. We incorporated these methods into SALT and HTML and conducted comparative user studies based on the MiPad scenarios. The experimental results show the system can gracefully cope with spontaneous speech and the users prefer the highly interactive nature of such systems even though there are no significant differences in the task completion rate and the understanding accuracy. However, the interactive interface does allow a more effective visual prompting strategy that contributes to the significantly lower out of grammar utterances.

1. INTRODUCTION

Just like automatic speech recognition (ASR), the problem of speech understanding can be tackled in a pattern recognition framework. While ASR aims at producing the word string, the pattern to be recognized for a speech understanding system is a semantic representation that captures the meaning of the utterance. More succinctly, given an utterance x and assuming a uniform cost, the problem of speech understanding can be viewed as to seek

$$\hat{S} = \arg\max_S P(S \mid x) = \arg\max_S P(x \mid S)P(S)$$

Usually, it is practical to first transcribe the speech waveform x into some linguistic units (e.g. word or phrase) so that the semantic extraction process is more tractable, namely,

$$\hat{S} = \arg\max_S \sum_W P(x \mid w, S)P(w \mid S)P(S) \quad (1)$$

Although the acoustic model $P(x \mid w, S)$ in a dialog application can be dynamically adjusted based on the dialog context, many speech understanding applications assume that the acoustic statistics are fully accounted for when the lexical items are given, i.e., $P(x \mid w, S) = P(x \mid w)$. Consequently, the semantic language model $P(w \mid S)$ and the decoding process realizing the argmax in the above equation become the key to forge a close link between the speech recognition and the understanding processes that insures the overall optimality of the system. In contrast to the recognition language model $P(w)$, a semantic language model $P(w \mid S)$ takes into consideration the semantic hypothesis S in predicting the word sequence, guiding the search process to favor outcomes more relevant to the understanding process.

As more and more applications are deployed, two factors emerge as playing a central role in affecting user's acceptance of a speech interface: robust understanding and response latency. By robust understanding we mean a system that can gracefully handle ill-formed utterances that are either phrased ungrammatically or contain disfluencies. Studies in this area, known as spontaneous speech, are abundant in the literature [1-6], with the majority of approaches adopting two-pass architecture: the system first recognizes the speech into a graph of lexical items that is consequently consumed by a second phase of semantic parsing to produce the semantic interpretation. Because the second pass starts after the utterance is processed, the two-pass architecture is ideal for systems using a turn taking interaction model where the user speaks commands in a user turn and the recognition and understanding processing takes place in the system turn. Advanced techniques can be employed to reduce the elapse time of a system turn without sacrificing the accuracy or robustness. On the other hand, the turn taking interaction model is not necessarily the only way to realize the full potential of speech in the computer human interaction as far as response latency is concerned. To fully take advantage of the temporal nature of speech, it is desirable that speech understanding can take place as immediately as possible.

In this paper, we report our efforts in creating a robust speech understanding system for highly interactive applications. The goal here is to carry out robust speech

understanding and generate proper responses back to the user in a speedy fashion, sometimes even before the user has finished speaking. We adopt a single pass approach that tightly combines the understanding with the recognition process, and extend the recognizer from performing the word synchronous recognition to what we call the *semantics synchronous understanding* (SSU). The temporal behavior of SSU does not change the nature of the understanding problem as described by (1). Rather, both the semantic language model and the decoding process undergo certain modification, which is discussed in greater detail in the following section. Undoubtedly, SSU represents a significant departure to the majority of speech applications and its efficacy remains unproven. As a start to assess its impacts, we conducted comparative studies with the MiPad scenarios that we previously reported [7]. The implementation of MiPad under the SSU framework and the experimental results are reported in Sec. 3.

2. SEMANTICS SYNCHRONOUS UNDERSTANDING

The interplay of three components jointly contributes to the effect of SSU. To incorporate semantic structure into the semantic language model $P(w \mid S)$, we employ the unified language model (ULM) technique that combines probabilistic context free grammar (PCFG) with N-gram. The decoding process therefore must be enhanced to navigate smoothly between the PCFG and the N-gram search spaces in a frame synchronous manner. Finally, to insure robustness in understanding, a sequential detection technique is used to accept or reject hypotheses in a timely fashion.

2.1. Unified language model for understanding

As PCFG and N-gram have complementary strengths and weaknesses, there have been many attempts to combine these two language modeling approaches into a single cohesive one, called the unified language model (ULM). The means of combination, however, comes with two flavors. One approach, aiming at speech recognition, extends the notion of a class in the class N-gram from being simply a list of words to phrases that are modeled by PCFG [8-12]. This technique leads to a more meaningful N-gram probability. Consider for example the sentence "*stock rose a quarter and a half to one hundred*." A regular trigram would compute probabilities such as $P(a \mid quarter, and)$, $P(one \mid half, to)$. In contrast, a ULM can treat the sentence as "*stock rose* **Percentage** *to* **Number**" and consider the probabilities $P($**Percentage** $\mid$ *stock, rose*$)$ and $P($**Number** $\mid$ Percentage, *to*$)$ instead. Such ULMs are thus capable of reducing the perplexity for the

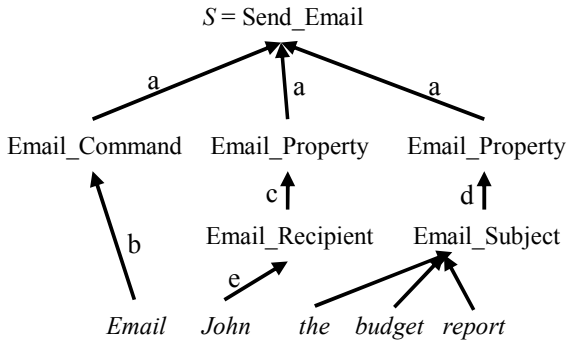

Fig. 1. Applying the ULM for understanding to a sentence with the semantic structure modeled by PCFG rules and some lexical elements, such as "*the budget report*", by N-gram.

recognition language model, but are not necessarily designed to uncover the semantic structure of a sentence.

In contrast, the second method, first explored by [13] and [14], embeds N-gram into PCFG in a way that the PCFG non-terminals are modeled by N-gram. With this modeling approach, one can trace the rules invoked in the PCFG to uncover the sentence structure. If the rules are authored to reflect the constructs of semantic objects, the result is a tree that represents the meaning of the sentence. We call this type of combination *ULM for understanding*. While PCFG alone has long been used for semantic extraction, the ULM is particularly useful to widen the lexical coverage due to the flexibility of N-gram.

Fig. 1 illustrates the semantic parse of the sentence w="*Email John the budget report*." The PCFG rules leading S to the "send email" hypothesis, as labeled in Fig. 1, are:

(a) $S \rightarrow$ Email_Command Email_Property*
(b) Email_Command → *Email*
(c) Email_Property → Email_Recipient
(d) Email_Property → Email_Subject
(e) Email_Recipient → *John*

In this example, only the pre-terminal Email_Subject is covered by N-gram, and $P(w \mid S)$ is the product of the N-gram for $P($*the budget report* $\mid$ Email_Subject$)$ and the probabilities of all the PCFG rules listed above.

More generally, N-gram can also be applied to model every right hand side expression of each rule in a ULM for understanding, not just the pre-terminals as in the example of Fig. 1. As shown in [15], the expectation maximization (EM) algorithm can be formulated to estimate the probabilities when sufficient amount of training data are available. However, a practical benefit of the ULM is that the model can be bootstrapped by smoothing a generic N-gram and manually crafted PCFG

rules with uniformly distributed probabilities before any field realistic data are obtained. Such a "bootstrap" model then can be employed in a prototype that doubles as a data collection tool for further model refinements. Consider again the email task in Fig. 1 as an example. While the potential names of email recipients can be dynamically compiled from the application database and therefore can be effectively modeled by PCFG, the wordings for the users to issue the email command can vary significantly. With enough data, it is reasonable to expect that replacing the right hand side of rule (b) with an N-gram can lead to a better lexical coverage.

2.2 Semantic object synchronous (SOS) decoding

A motivation of applying the same framework of speech recognition to understanding, as described in (1), is to apply the advancements made during the past decades in the recognition area to speech understanding. One such advancement is the frame synchronous decoding method for continuous speech recognition [16]. It provides an elegant technique for the speech recognizer to accumulate likelihood scores for various hypotheses synchronously to the incoming audio. Commercial speech products, such as [13], incorporate this technique in either PCFG or N-gram applications to provide speedy responses. The technique is therefore very relevant here as we are pursuing the same goal for speech understanding.

A challenge of applying the frame synchronous decoding technique to ULM for understanding is to mediate the search spaces of the PCFG and the N-gram, namely, how a hypothesis can navigate back and forth from PCF to N-gram and vice versa. Since ULM for understanding is basically a PCFG with embedded N-gram, the approach we take can be explained by considering the following simple production

```
LCFG N-gram RCFG
```

where the LCFG and RCFG denote the left and right context of the embedded N-gram. While migrating from LCFG into the N-gram, the search process treats the N-gram just like a non-terminal and applies the normal transition probabilities. Inside the N-gram, however, the word sequence probabilities are computed by constantly considering a backoff to the PCFG alongside with staying within the N-gram, namely,

$$P(w_n \mid w_{n-1}, w_{n-2}, \ldots) = \begin{cases} \lambda P(w_n \mid Ngram, w_{n-1}, w_{n-2}, \ldots) \\ (1-\lambda) P(w_n \mid RCFG) P(RCFG \mid w_{n-1}, w_{n-2}, \ldots) \end{cases} \quad (2)$$

where λ is the N-gram weight over PCFG, and $P(RCFG \mid w_{n-1}, \ldots)$ the back-off probability of the N-gram. Note that the second part of (2) hypothesizes the scenario in which the search path should migrate back to PCFG by treating w_n as an out of vocabulary word for the N-gram. Also, since (2) can be applied successively to each word, the semantic language model scores can be updated in a word synchronous manner. Similarly, since all the semantic objects are modeled in ULM as PCFG non-terminals (Sec. 2.1), the search process leads to a scoring mechanism that is synchronous to the semantic object instantiation.

2.3 Hypothesis testing with sequential detection

As the semantic objects are instantiated and semantic hypotheses are generated word synchronously, a question arises as to when is the right time to accept or reject the hypothesis and execute the domain logic associated with the semantic objects. Obviously, the onset of a hypothesis is not a good candidate because it may lead to too many false alarms. However, waiting for too long increases the latency of the process, which is also undesirable. A mathematical framework, known as the sequential detection [17], addresses this very issue.

The problem of sequential detection can be stated as follows: given a continuing observation $x = (x_1, x_2, \ldots, x_t, \ldots)$, the goal of sequential detection is to determine that at any particular time, whether the collected observations are sufficient to accept or reject an event hypothesis H, or such decision should be deferred. It can be shown [17] that the optimal decision (in terms of the shortest time to decision) is the sequential probability ratio (SPR) test:

$$SPR = \frac{P(x_1, \ldots, x_t \mid H)}{P(x_1, \ldots, x_t \mid H')} : \begin{cases} > A & \text{accept} \\ < B & \text{reject} \\ \text{otherwise} & \text{defer} \end{cases} \quad (3)$$

where A and B are two decision thresholds, and H' denotes the anti-hypothesis of H. Common to the detection systems, two types of errors, false detection and false rejection, can be made whenever there is a decision. Let P_F and P_M denote the probabilities of these two types of errors, respectively. It can be shown [16] that the choices of the two decision thresholds are bounded by

$$A \le \frac{1-P_M}{P_F}, \quad B \ge \frac{P_M}{1-P_F}. \quad (4)$$

Note that the error rates can be arbitrarily reduced by picking the thresholds as described in (4), and the choice of the thresholds does not depend on the statistic properties of the signal. However, these factors directly impact the latency to decision which, in our study, plays a critical role in the user experience (Sec. 3).

Although in the typical sequential detection problem it is theoretically possible to defer the decision indefinitely, a time limit can usually be administered in practice. This amounts to force a rejection or detection decision in (3) when the number of observations has reached a prescribed upper bound. We apply such a "bounded" sequential

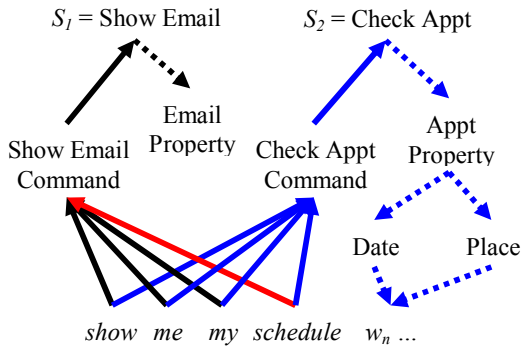

Fig. 2. An illustration of two competing semantic hypotheses being composed as words are recognized under the SSU framework. Dotted lines represent semantic objects and phrases predicted by the semantic language model.

detection in SSU to determine the timing of accepting or rejecting a hypothesis. Following the notation of (1), we use $P(x \mid S)P(S)$ as the base for SPR test, and the L^P norm of all the competing hypotheses as the measure of the anti-hypothesis, i.e.,

$$\mathrm{SPR}(S) = \frac{P(x \mid S)P(S)}{\sqrt[p]{\sum_{S' \neq S} \left[P(x \mid S')P(S')\right]^p}}.$$

It is intuitive to consider a special case where L^∞ norm is employed, namely, each hypothesis is always compared against the strongest competitor. Note that such choice guarantees that, at any given time, there is at most one leading hypothesis whose SPR > 1, although it is quite likely that at the onset of the search process many hypotheses can tie at SPR = 1.

Fig. 2 illustrates how the semantic language model, the semantic object synchronous decoding, and the sequential detection are combined under the SSU framework using the sentence "*show me my schedule*" as an example. The first three words are shared by the Show Email and Check Appointment commands. When they are hypothesized by the speech recognizer, the semantic language model guides the decoder to construct the corresponding parses and predict the semantic objects that are likely to follow. For the Send Email hypothesis, the ULM predicts email related properties, such as the sender or email receiving date. For the Check Appointment hypothesis, they are the date, time, place, subject, and other properties of an appointment (Fig. 2). As the user utterance continues, the SOS decoder keeps track of the SPRs for all the hypotheses, pruning those under the rejection threshold as appropriate. When the SPR of a leading hypothesis reaches the detection threshold, the system commits itself to the hypothesis and prunes all its competitors. In the example of Fig. 2, this is likely to happen when the word "*schedule*" is recognized and a significant difference in the scores is induced. As a result, SSU will report the detection and enter a new sequential detection cycle for the rest of the utterance with a bias towards the semantic objects of the Check Appointment task. If the following word sequence constitutes a date expression, for example, SSU will regard it as a date for an appointment, rather than the date an email is received or sent.

2.4. Coping with spontaneous speech in SSU

In field applications, SSU appears to exhibit some robust understanding properties in coping with non grammatical sentences and phenomena of spontaneous speech. These properties can be attributed to the following factors in the SSU design. First, the language model for the speech recognition is a semantically structured model that can accommodate large syntactical variations in a sentence. The semantic structure can be modeled with PCFG rather straightforwardly, and the accompanied N-gram in ULM allows extraneous lexical items to occur in a sentence that can be subsequently wielded out by the SPR test. Most importantly, SSU can accept self corrections as the utterance is processed in an ongoing basis. Consider the example in Fig. 2 with the sentence *"show me my schedule for Monday… I mean Tuesday."* When *"Monday"* is recognized, SSU interprets the user's intention as to bring up Monday's calendar based on the operations described in Sec. 2.3. As the processing continues to *"Tuesday,"* SSU will re-commit the Date semantic object and change the calendar view accordingly. In an interactive user interface, SSU does not seem to have to distinguish whether the user has indeed issued a correction, or has intended to execute multiple commands as viewing the calendar of multiple days in the first place.

Advanced understanding system often has to resolve meaning not just at the utterance but at the discourse level. A SSU system can be used in the discourse context. The discourse semantic processing and reference resolution algorithms described in [18], for instance, can be applied whenever SSU commits a semantic object (Sec. 2.3). Accordingly, SSU can be applied to a multimodal environment, if the multimodal integration is treated as a discourse level reference resolution problem, as in the MiPad application described in the following section.

3. EXPERIMENTS AND RESULTS

To assess the efficacy of SSU, we implemented two prototypes using the MiPad scenarios [7] and the bootstrap language model (Sec. 2.1) in HTML and the speech application language tags (SALT) [19,20].

3.1 Experiment design

MiPad is a mobile personal data assistant that allows users to manage the contact list, send or read email, create or check calendar items through a graphical user interface (GUI) with a stylus or speech. The user interaction in the previous work [7] is based on the turn taking model commonly used in the speech interface design. A user turn begins when the user holds the stylus in an input field and starts speaking. It ends when the user lifts the stylus, at which time the system stops collecting audio and starts speech processing and response generation. MiPad has a generic input field always available for sophisticated speech commands that accomplish many GUI steps in one utterance. Although the semantic language model is still applicable, the understanding system does not have to perform a causal processing as in the case of SSU. The turn taking MiPad prototype utilizes the so-called "single" recognition mode in SALT that instructs the recognizer to utilize the speech end points to produce the best recognition results. We use the turn taking MiPad as a control to study the SSU MiPad that departs from the turn taking model and presents partial responses as soon as it can, often while the user is still talking. Because many speech commands can be lumped together under SSU, the user has roughly an open microphone experience. Furthermore, the user does no longer have to hold onto a particular input field to provide contextual information. Such information is predicted from the domain via semantic language model (Sec. 2) in the SSU MiPad. As a result, the stylus in SSU MiPad is freed up for pen gesture that can be overlaid on top of speech inputs. Other than these differences, the two MiPad prototypes share the identical look and feel in GUI and application domain logic, and their language models have comparable perplexities.

Six members of Microsoft .NET speech group were recruited for user studies. All of the subjects, including both native and non native English speakers, are at least casual speech users and are familiar with the current speech technology. Each subject was asked to complete numerous tasks, such as arranging a new meeting or check email, using either the turn taking or the SSU MiPad. The name lists in the MiPad task varied slightly among subjects, so as the language model perplexities as a result. On the average, each subject spent 30 minutes in total on various experimental sessions. The default settings for a SALT implementation by Microsoft, freely available as an add-on to the Internet Explorer, were used for all the speech parameters such as the N-gram, language model weight, rejection threshold and the speaker independent acoustic model. The studies were conducted on a Tablet PC with its built-in microphone in an office where hard disk, fan, and other environmental noises can be clearly heard in the recordings.

The recordings were transcribed into text as well as semantic objects. We measure the understanding accuracy in the same manner as the recognition accuracy, i.e., it is the percentage of the correctly identified semantic objects minus the insertion, deletion and substitution errors. Note that we label each spontaneous or out of domain speech fragment as an OOD semantic object, which is counted as correctly understood if it is rejected by the system. We are lenient in treating semantically confusable phrases. For example, it is counted as one correctly recognized semantic object when the time expression "*from ten to eleven*" is interpreted as a starting time at 10:50, even though the user's intention is really to specify the duration from 10:00 to 11:00. Together with the correcting utterance "*I mean from ten o'clock to eleven o'clock*" that follows, our transcription rules will regard the user has issued three semantic objects, first the start time 10:50, then the corrected start time 10:00 and the end time 11:00.

3.2 Experimental Results

For the dictation task in MiPad, the subjects were asked to read a 57-word paragraph of a call for papers in an IEEE Computer Magazine. The word accuracy varies quite considerably among the individuals, ranging from 37% to 89%. In contrast, the understanding accuracy ranges from 67% to 95%, mostly due to a much lower perplexity in the ULM. The majority of the understanding errors stem from rejecting the in-domain phrases, such as proper names, which are counted as the deletion errors.

There do not appear discernible differences in terms of understanding accuracy between the turn taking and the SSU MiPad, although a direct comparison is difficult. This is because the users adjust quickly to the interfaces and exhibit significantly different speech usage patterns. The OOD rate for SSU MiPad is much lower. Normalized with Student's *t*-distribution, the OOD rate for the SSU MiPad is 2.5 standard deviations lower than its counterpart ($df = 5$). The higher OOD rate in the generic input field of the turn taking MiPad results in more rejections for long utterances, effectively leading the users to fall back to GUI for navigation and application controls most of the time. As a result, the speech usages are often narrowly directed at individual input fields and shorter. In contrast, the SSU MiPad allows users to follow visual cues to adjust their wordings dynamically for better recognition, using a user interface design principle called "what you see is what you can say" (WYSIWYCS) [21]. As a result, speech is used more often and the utterances tend to be longer and more sophisticated.

Although the subjects are all aware of the multimodal capabilities, they tend to use speech as the first choice to

correct speech errors, switching to the alternative pen modality only if repeated attempts fail. The observation is consistent with the findings of [22]. Unexpectedly, we observe that almost all users slow the speaking rate and over-articulate during correction, including those who fully understand that such a speaking style would further the signal from the fluent speech under which the recognizer is trained and therefore degrade its accuracy. Many users were unaware and even surprised at the change of speaking styles during the post experimental interviews.

At current settings, it can take up to a few seconds for the SSU MiPad to display understanding results on the screen. The bulk of the latency occurs at the sequential detection, probably due to a poor $P(x \mid S)$ measurement. Almost all users indicate this is the area for improvement. Nevertheless, all subjects except one feel the SSU interface palatable and, for some, faster to accomplish the tasks. The speed perception, however, is not supported by data as both interfaces are statistically tied in terms of the task completion rate.

4. SUMMARY

A spoken language understanding system capable of generating immediate feedbacks to the user is presented in the article. The system utilizes ULM and SOS decoding to integrate speech understanding with recognition into a single pass processing. A sequential detection algorithm based on SPR test is devised to determine the timing of rejecting or accepting hypotheses. Experiments based on MiPad seem to suggest that such a system possesses basic robust characteristics in coping with spontaneous speech. Although the interaction model is considerably different from many existing spoken language interfaces, human subjects view it as acceptable and many, attractive.

ACKNOWLEDGEMENT

Fil Alleva and his team first implemented (2) in [13] as an improvement to the limited domain grammar in SAPI 4.0.

REFERENCES

[1] Seneff S., "Robust Parsing for spoken language systems", in *Proc. ICASSP-92,* San Francisco, CA, 1992.

[2] Antoine J.Y., Caillaud B., Caelen J., "Automatic adaptive understanding of spoken language by cooperation of syntactic parsing and semantic priming", in *Proc. ICSLP-94*, Yokohama, Japan, 1994.

[3] Issar S. and Ward W., "CMU's robust spoken language understanding system", in *Proc. Eurospeech-93*, Berlin, Germany, 1993.

[4] Wang Y., "A robust parser for spoken language understanding", in *Proc. Eurospeech-99*, Budapest, Hungary, 1999.

[5] Kaiser E., Johnston M., Heeman P., "Profer: Predictive robust finite state parsing for spoken language", in *Proc. ICASSP-99*, Phoenix, NM, 1999.

[6] Kawahara T., Lee C. H., and B. H. Juang, "Flexible speech understanding based on combined key-phrase detection and verification," IEEE Trans. Speech and Audio Processing, Vol. 6, No. 6, pp. 558-568, November 1998.

[7] Huang X. *et al*, "MiPad: A next generation PDA prototype", *Proc. ICASSP-2001*, Salt Lake City, UT, 2001.

[8] Gillett J. and Ward W., "A languge model combining trigrams and stochastic context free grammars", in *Proc. ICSLP-98*, Sydney, Australia, 1998.

[9] Niemann H., Eckert W., Gallwitz F., "Combining stochastic and linguistic language models for recognition of spontaneous speech", in *Proc. ICASSP-96*, Atlanta, GA, 1996.

[10] Nasr A., Esteve Y., Bechet F., Spriet T., De Mori R., "A language model combining N-grams and stochastic finite state automata", in *Proc. EuroSpeech-99*, Budapest, Hungary, 1999.

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

[12] Benedi J., Sanchez J., "Combination of N-grams and stochastic context free grammars for language modeling", in *Proc. Coling-2000*, Saarbrucken, Germany, 2000.

[13] Microsoft Speech Application Interface (SAPI) Version 5.0, 1999.

[14] Hacioglu K., Ward W., "Dialog-context dependent language modeling combining N-grams and stochastic context free grammars", in *Proc. ICASSP-2001*, Salt Lake City, UT, 2001.

[15] Wang Y., Acero A., "Concept acquisition in example-based grammar authoring", in *Proc. ICASSP-2003*, Hong Kong, China, 2003.

[16] Ney H., Ortmanns S., "Dynamic programming search for continuous speech recognition," *IEEE Signal Processing Magazine*, pp. 64-83, 1999.

[17] Poor H. V., An introduction to signal detection and estimation, Springer-Verlag, New York, 1988.

[18] Wang K., "A plan-based dialog system with probabilistic inferences", in *Proc. ICSLP-2000*, Beijing, China, 2000.

[19] Wang K., "SALT: a spoken language interface for Web based multimodal dialog systems", in *Proc. ICSLP-2002*, Denver, CO, 2002.

[20] Wang K., "Semantic object synchronous decoding in SALT for highly interactive speech interface", in *Proc. Eurospeech-2003*, Geneva, Switzerland, 2003.

[21] Wang K., "A study of semantics synchronous understanding for speech interface design," in *Proc. ACM Symposium UIST-2003*, Vancouver, Canada, 2003.

[22] Suhm B., Meyers B., Waibel A., "Multimodal error correction for speech interface", ACM Trans. on CHI, Vol. 8, No. 1, pp. 60-98, March 2001.