Flexible Speech Understanding Based on Combined Key-Phrase Detection and Verification

Tatsuya Kawahara, *Member, IEEE,* Chin-Hui Lee, *Fellow, IEEE,* and Biing-Hwang Juang, *Fellow, IEEE*

***Abstract*— We propose a novel speech understanding strategy based on combined detection and verification of semantically tagged key-phrases in spontaneous spoken utterances. Key-phrases are defined in a top-down manner so as to constitute semantic slots. Their detection directly leads to robust understanding. A phrase network realizes both a wide coverage and a reasonable constraint for detection. A subword-based verifier is then incorporated to reduce false alarms in detection and attach confidence measures of the detected phrases. This set of phrase confidence measures, when incorporated in a spoken dialogue system, forms a basis for designing intelligent speech interfaces that accept only verified key-phrases and reprompt users to clarify unspecified or unrecognized portions. Several forms of confidence measures based on subword-level tests are investigated. The proposed approach was tested on field data collected from real-world trial applications. The combined detection and verification strategy drastically improves the accuracy in handling out-of-grammar utterances over the conventional decoding approaches while maintaining the performance for in-grammar utterances.**

***Index Terms*—Dialogue systems, key-phrase detection, speech recognition, speech understanding, utterance verification.**

I. Introduction

In RECENT years, several spoken dialogue systems based on continuous speech recognition have been evaluated in real-world applications. These systems use deterministic finite state grammars to accept and decode typical user utterances, because there are no data available to train statistical language models for specific tasks. The use of a rigid grammar represented by a finite state machine is reasonably effective for typical *in-grammar* sentence patterns, i.e., sentences that can be described by the finite state grammar. However, in real-world environments, we have observed wide utterance variation inherent in a large user population that is not covered by the task grammars, even though they had been tuned manually by system developers during the trial period. In addition to the desired information, these samples usually include extraneous words, hesitations, repetitions, disfluency and other unexpected expressions [1]. Most of such utterances contain some key-phrases that are task-related and may be sufficient for partial or full understanding. Other samples are not relevant to the task and should be rejected.

Manuscript received December 18, 1996; revised January 29, 1998. This work was performed while T. Kawahara was visiting Bell Laboratories in 1995–1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Picone.

T. Kawahara is with the School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: kawahara@kuis.kyoto-u.ac.jp).

C.-H. Lee and B.-H. Juang are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974-0636 USA.

Publisher Item Identifier S 1063-6676(98)07792-X.

When we review most of the spoken dialogue systems, their task specifications are highly well-defined, so that necessary information for the system is described with a definite set of task-related slots. Their typical examples include form filling or information retrieval by voice. Therefore, speech understanding problem can be formulated as extracting or detecting the task-related slots from unconstrained utterances. These slots are usually defined with keywords or key-phrases such as time and place. One of the design goals of a *flexible speech understanding* system should be to detect the semantically significant portions and reject the *out-of-grammar* and *out-of-task* portions of the input utterance. Utterance verification technique enhances this property by giving confidence measures to recognition results. Combined with a flexible dialogue manager, the detection and verification framework will realize partial understanding and disambiguation of unclear portions through the subsequent dialogue session.

One of the most comprehensive projects on spoken dialogue processing so far was the Air Travel Information System (ATIS) project sponsored by ARPA [2]. In such a task where a lot of data have been collected, the use of a statistical language model (n-gram) is typical and can be effective. Moreover, statistical concept modeling [3], [4] has been studied and demonstrated to be a viable way to model semantics in domain-restrictive tasks. In actual situations, however, it is not realistic to assume that a large amount of dialogue data are available for training such models for every single application. The effort in data collection and labeling is often expensive, labor intensive, and the results are potentially error-prone and sometimes undesirable. Thus, the prevailing statistical language modeling in the ATIS evaluation cannot be applied directly to many of the real-world applications.

Therefore, most of the real-world dialogue systems use finite state grammars for the specific tasks. The recognizer tries to match or decode the whole utterance input into possible word sequences accepted by the grammar. Usually the grammar should realize both a wide coverage to accept a variety of sentences and a small perplexity to achieve high recognition performance. These requirements become very difficult to satisfy when a wide variety of spontaneous utterances need to be coped with in real-world environments. For example, in an apparently simple subtask of recognizing responses to the prompt "What is your drop-off date?" in a car reservation task, some users include unexpected phrases such as "I will be returning the car on September fourth please" in their answers. Many utterances contain hesitation like "August fit August fifth." The situation becomes even worse when the task

involves more complex queries. Tuning the task grammars to cover all possibilities would be an endless effort. The problem originates from the framework of decoding that assumes a rigid sentence-level grammar and applies the uniform constraint on the whole input. The inclusion of filler models in the definition of a finite state grammar works for limited samples that closely follow the rigid grammar. But it does not solve the problem fundamentally.

As a more robust strategy, word spotting approaches [5], [6] have been studied. They are classified into two approaches in terms of the modeling of non-keyword parts. The first is the use of a large vocabulary continuous speech recognition (LVCSR) system (e.g. [7], [8]). It attempts to incorporate as much lexical and pragmatic knowledge as possible. However, it does not model the ill-formed phenomena such as hesitations and repairs, which are often found in spontaneous speech. The approach based on LVCSR is also not a realistic solution both in performance and efficiency, especially in cases where the possible vocabulary is not well specified or the statistical language model for the subtask is not reliably trained. The second word spotting approach is to use a general acoustic sink model (e.g. [9]) or a parallel network of context-independent phone models (e.g., [10]). However, such simple models are usually not sufficient to characterize non-keyword events especially when the size of keyword vocabulary is over a few dozen. The keyword models are easily matched with the irrelevant portions, causing so many false alarms that cannot be easily handled with subsequent processing. Most of the past works tune the keyword models and the sink model in a vocabulary-dependent manner (e.g., [10]), sacrificing the advantage of subword-based recognition. While whole-word-based keyword spotting is possible, the approach has only proven effective in very small vocabulary tasks (e.g., [9]).

In this paper, we propose a *combined detection and verification* approach that realizes flexible speech understanding. We first extend the conventional keyword spotting framework to key-phrase detection. It is well known that longer speech units such as phrases are more stable than words for spotting even when they are embedded in extraneous speech. Key-phrases are also semantic units that represent partial task-related meanings in a sentence.

The idea of extracting such semantic units from a complex sentence is consistent with the similar findings about partial parsing in the ATIS project. Several *template matching* algorithms [11], [12] and *robust parsing* algorithms [13], [14] oriented toward parsing ill-formed sentence fragments were found quite effective in handling some disfluencies in the ATIS task. Most of these approaches, however, assume that a word sequence (text) has been obtained by some speech recognizer (using N-gram models). It is very difficult to realize effective postprocessing with the current LVCSR systems unless a large word lattice with focus on keyword and key-phrases is generated. We need precise N-best list for key-phrases with acoustic confidence, but we can merge non-keywords as garbage. A simple deep word lattice will generate too many irrelevant hypotheses for speech understanding.

Our strategy detects such key-phrases directly from speech and performs optimization jointly with the semantic constraints. It can be viewed that the detection module proposes possible theories for the system to explore in subsequent processing. Since many theories are still likely as the result of partial matching, a key-phrase verification module is incorporated to select reliable theories and eliminate false alarms. After this preliminary hypothesis pruning, the remaining theories are parsed and merged to form valid sentences as well as their semantic frame representations.

In real-world spoken dialogue processing problems where the definition of the task and the vocabulary is always evolving, portability of the system is significant. In some tasks such as making inquiries on movie titles, the vocabulary of movie titles changes regularly. Therefore, not only acoustic models for recognition but also verification formulation should be vocabulary-independent subword-based. Moreover, the language model has to be portable, since writing rigid sentence grammars takes much human effort and training statistical models needs huge data collection. Specifying keywords or key-phrases is much easier for system designers, as they can often be automatically derived from the task specifications. The set of key-phrases will accept a wider variety of utterances than sentence grammars can. Especially in dialogue-based systems, it is possible to set up subtask grammars according to the dialogue state and apply them to a large-scale task.

The rest of the paper is organized as follows. We first present an overview of the proposed detection and verification system in Section II. Key-phrase detection and key-phrase verification are described in detail in Sections III and IV, respectively. Issues related to sentence parsing and verification are discussed in Section V. Experimental results on several subtasks are reported in Section VI. Finally we summarize our findings in Section VII.

II. Detection and Verification Strategy

It is becoming increasingly clear that an automatic speech recognition system needs to have both high accuracy and a friendly interface that allows a user to speak naturally and spontaneously without imposing a rigid format. Our strategy for handling such spontaneous utterances, particularly when contemplating domain-specific services, is to focus on a finite set of vocabulary words most relevant to the intended task and make use of the technology of *utterance verification* (UV). The system then detects and identifies the in-vocabulary keywords and key-phrases that may be embedded in the fluent speech utterance, while rejecting irrelevant portions.

The simple word spotting scheme that uses small templates can be easily triggered by local noise or confusing sounds. Using a longer unit is advantageous because it can incorporate more distinctive information and realize stable acoustic matching both in recognition and in detection. Therefore, one major feature in our strategy is to use key-phrases as the detection unit in addition to using keywords. A key-phrase consists of one or a few keywords and functional words. For example, “in the morning” for a time period, and “in downtown Chicago” for a local area. In most situations, they are uttered without a break even in spontaneous speech. Furthermore, they are tagged with conceptual information. In fact, we define our key-phrases so as to correspond to semantic slots such as time

and place. Unlike bottom-up phrases defined by the N-gram scheme [15]–[17], our top-down phrases are directly mapped into semantic representations. Thus, detection of them directly leads to robust understanding.

The other main feature is to incorporate utterance verification technique to realize ideal detection mechanism that does not match irrelevant portions of speech without using large-vocabulary non-keyword knowledge. One of the most significant problems in the conventional recognizers is that they do not know how confident their outputs are. Therefore, we have been studying utterance verification methods that perform hypothesis tests on the recognized results and give them confidence measures [17]–[20]. Based on the confidence measures, the system can reject utterances that contain superfluous acoustic events such as out-of-vocabulary words, any form of disfluency and ambient sounds, as well as invalid inputs that have no key-phrases. In this work, we integrate the verification technique into detection in order to select reliable detection and eliminate improper matching or false alarms. The detected key-phrase hypotheses are passed into verification module for validation.

The keyword or key-phrase verification is different from the conventional utterance verification, because it is not the final decision. False rejection of correct hypotheses is critical, while accepted false alarms can still be eliminated in the subsequent sentence parsing and verification process. Furthermore, since verification of phrases is done with partial input of fewer subword segments than the whole utterance verification, it demands more reliable confidence measures.

Finally, in order to understand the whole utterance, we perform sentence-level processing that combines detected key-phrases and verifies the end result.

A. Overview of the System

Thus, our overall strategy consists of the following steps, as depicted in Fig. 1.

1) *Key-phrase detection*: A set of key-phrases are detected using a set of phrase subgrammars specific to the system prompt in the dialogue. The key-phrases are labeled with semantic tags, which are useful in sentence-level parsing.
2) *Key-phrase verification*: The detected key-phrases are verified and assigned confidence measures. The process attempts to eliminate false alarms. It is a combination of subword-level verifications that use *anti-subword models* to test the individual subwords of the recognized results.
3) *Sentence parsing*: The verified key-phrase candidates are connected into sentence hypotheses using task-specific semantic knowledge. A stack decoder is used to search for the optimal hypotheses that satisfy the semantic constraints [21].
4) *Sentence verification*: The best sentence hypotheses are verified both acoustically and semantically for the final output.

The framework will realize not only flexible understanding but also portable and general one, that is vocabulary-independent and even task-independent.

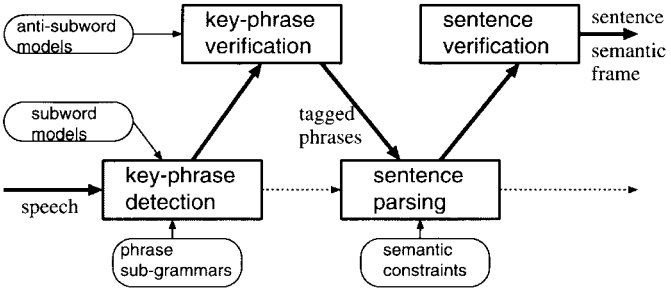

Fig. 1. Outline of the strategy.

For vocabulary-independent recognition, universal context-dependent subword units are selected and trained without influence of a specific vocabulary set. The verification is also formulated in a subword-based manner. Both phrase verification and sentence verification are carried out by combining likelihood ratio scores of constituting subwords. Moreover, phrase subgrammars are also easily constructed by specifying the values of semantic slots that the system tries to extract. Thus, system designers will not have to precisely predict what kinds of expressions are used including filler phrases and extraneous words.

B. Baseline System and Task-Independent Acoustic Modeling

The baseline system used for training and recognition is described in detail in [22]. Input speech, sampled at 8 kHz, is initially preemphasized $(1 - 0.95z^{-1})$ and grouped into frames of 240 samples with a shift of 80 samples. For each frame, a Hamming window is applied followed by a tenth-order linear predictive coefficient (LPC) analysis. A liftered 12-dimensional LPC-derived cepstral vector is then computed. The first and second time derivatives of the cepstrum are also computed. Besides the cepstral-based features, the log-scaled energy normalized by the peak and its first and second order time derivatives are also computed. Thus, each speech frame is represented by a vector of 39 features.

The lexical representation of each vocabulary entry is automatically generated using the Bell Labs Text-to-Speech grapheme-to-phoneme transcription rules. No hand-tuning is performed. Recognition is accomplished by a frame synchronous beam search algorithm [22] to determine the sequence of words that maximizes the likelihood of the given utterance. A forward-backward N-best search algorithm [23] is also used to generate multiple word string candidates.

Instead of designing a task-dependent speech recognition system that only works well for a particular task, we aim at having a system that works well for a wide range of tasks without re-training acoustic phone models for each new task. One way to accomplish this is through discriminative training of task-independent (TIND) phone models. The reader is referred to [24] for an in-depth discussion of TIND training.

The data base used for task-independent (TIND) training is a set of 12 000 utterances of general phrases of American English collected by AT&T[1] over long distance public subscribers telephone network (PSTN). More than 2,000 talkers

[1] The data base was designed and recorded by R. Sachs of the AT&T Voice and Audio Processing Architecture Department, Holmdel, NJ, in 1993.

each speaking up to seven phrases were included. Each phrase was semantically correct with length ranging from two to four words. The selection of the phrases was based on a greedy algorithm such that a maximum triphone coverage is obtained.[2] Over 6,000 distinct words were included in the recording.

In order to broaden the context coverage to deal with all unknown tasks and maintain the performance advantage of the context-dependent (CD) units over context-independent (CI) units in all experiments, we use the set of right CD (RCD) phone units as a universal TIND phone set [24]. Since not all single-context CD phone units appear in the training set and not all units appear frequently enough, we used a threshold of 50 to limit the choice of the number of single-context units. This resulted in a set of 1034 right CD units (as opposed to the full set of 1640 units). We also supplemented it with the set of 41 CI units to handle possible missing context due to the above unit reduction rule. This gives a set of 1075 RCD+CI phone units.

Except for the background silence unit, each subword unit is modeled by a three-state left-to-right hidden Markov model (HMM) with no state skip. Each state is characterized by a mixture Gaussian state observation density. A maximum of eight mixture components per state is used. Training was done with an iterative segmental ML algorithm (e.g., [22]) in which all utterances were first segmented into subword units. The Baum–Welch algorithm was then used to estimate the parameters of the mixture Gaussian densities for all states of subword HMM's. The HMM parameters were then refined using the segmental *generalized probabilistic descent* (GPD) algorithm to minimize phone recognition error [25]. It was observed that such a training procedure attempts to maximize the separation between phone models and gives a better recognition performance than the ML-trained models. It also achieves the goal of TIND training without taking into account of vocabulary and grammar specification of new tasks [24].

III. Key-Phrase Detection

For each subtask, key-phrase patterns are described as a finite state grammar. Since the set of keywords and key-phrases are to be directly mapped to semantic values of task-related slots, they are easily derived from the task definition. For example, in a subtask of asking for a date, possible words that can fill the date slot are derived. Such phrases are defined to include functional words or patterns like "at the" or "near" instead of using keywords alone. It was demonstrated in [21] and confirmed in some of the experiments here that this syntactic constraint enables more stable matching and improves detection accuracy. We also define filler phrases that are not covered by any of the key-phrases but often accompany the key-phrases. In this paper, however, we use only minimal filler phrases known *a priori* and do not tune subgrammars, in order to demonstrate generality and portability of our approach.

A. Key-Phrase Network

The key-phrase and filler phrase subgrammars are compiled into a finite state network, where key-phrases are recurrent and an acoustic sink model is embedded between key-phrase recurrences. Simple recurrence, however, causes ambiguity. For example, if we allow any repetitions of the days of the month, we cannot distinguish between "twenty four (24)" and "twenty (20)"+ "four (4)." Therefore, we incorporate constraints that inhibit impossible connections of key-phrases.

As a whole, the detection unit is a network of key-phrase subgrammar automata with their permissible connections and iterations. The constraint achieves wider coverage with modest perplexity than sentence-level grammars. It can be easily extended to a stochastic language model by estimating the connection weight.

The network characterizes a *semantic concept* of a specific subtask such as date and location. Furthermore when we construct a network that consists of parallel key-phrase networks representing all subtasks, a complex input utterance can be decoded as a sequence of semantic concepts without a strict syntactic constraint on whole sentence patterns.

B. Detection Algorithm

The detection algorithm adopted in this work is based on the forward-backward two-pass search [23], although a one-pass detection is possible. For the detection purpose, we incorporate hypothesis merging and pruning.

Although the A*-admissible stack-decoder can find the correct N-best hypotheses of word strings, the resulting N-best hypotheses are generally of similar word sequences with one or two replacements. Since our concern is to obtain key-phrase candidates on the partial input, not string hypotheses on the whole input, we abandon the (string) hypotheses whose further extension will lead to the same (phrase) sequence as the previously extended ones.

The merging and pruning mechanism is implemented by marking merging states of the key-phrase network. A merging state corresponds to the node where key-phrases or filler phrases are completed and further extension starts next new phrases. When a hypothesis popped by the stack-decoder is tagged as a complete phrase for output, it must be at some merging state of the grammar network. Then, we extend one more word and time-align the phrase with the best extension. If the grammar node was reached at the same time-point by any of the previous hypotheses, then we discard the current hypothesis after outputting the detected phrase. Otherwise, the time-point is marked for further search.

The detection algorithm is quite efficient without redundant hypothesis extensions. It suboptimally produces the correct N-best key-phrase candidates by the order of their scores. It terminates at the desired number of phrases or a certain score threshold. In the experiment described later, the detection is terminated when the score of the hypothesis gets lower than 0.99 times the best score.

IV. Phrase Verification and Confidence Measures

We adopt a vocabulary-independent approach for verification of detected phrases [26] so as to be applicable to a new subtasks with new vocabularies and grammar definitions. For the purpose, both the verifier training and verifier operation are

[2] The algorithm was graciously provided by J. van Santen of the Linguistics Research Department, Bell Labs, Murray Hill, NJ.

subword-based and independent of any specific task domains. The verifier is constructed for every subword and its training is performed with a phonetically balanced database that was used for training subword models. The verification procedure is a combination of subword-level hypothesis tests. Specifically, it consists of following three steps. First, detected phrase hypotheses are segmented into subword units. Next, hypothesis tests are performed for every subword segment. Then, the phrase verification is done by combining their results.

A. Subword-Level Acoustic Verification

For every subword n in a phrase sequence, a verification score is computed based on its corresponding *likelihood ratio* (LR) statistic, defined as

$$LR_n = \frac{P(O|H_0)}{P(O|H_1)} = \frac{P(O|\lambda_n^c)}{P(O|\lambda_n^a)} \tag{1}$$

where O is the observed speech segment, H_0 is the *null hypothesis* that subword unit n is present in the speech segment O, H_1 is the *alternative hypothesis* that subword n is not in the speech segment O, and λ_n^c and λ_n^a are the corresponding subword and *anti-subword* models for subword n, respectively [18]. The observation sequence O is aligned for subword n with the Viterbi algorithm as the result of recognition.

The anti-subword model can be considered as a model that approximately characterizes the alternative hypothesis H_1. For every subword model, a corresponding anti-subword model is trained specifically for the verification task. It is constructed by clustering the highly confusing subword classes [26]. It has the same structure, i.e., number of states and mixtures, as the correct subword HMM. The use of an anti-subword model as a reference is more discriminative than unconstrained decoding of subword models [26], because the anti-subword model is more sensitive to the similarity of subwords and free from the performance of subword-level recognition. In fact, it has the ability to reject substitution errors by the recognizer. Here, we use a context-independent anti-subword model, while the recognition is done with the context-dependent model.

By taking the logarithm of (1) and normalizing it by the duration (length) l_n of the speech segment O, we define LLR_n as,

$$LLR_n = \{\log P(O|\lambda_n^c) - \log P(O|\lambda_n^a)\}/l_n. \tag{2}$$

Since the first term of the equation is exactly the recognition score, we just offset the score by that computed with the anti-subword model.

B. Confidence Measures of Phrase Hypothesis

A confidence measure (CM) for phrase verification combines the subword-level verification scores. It can be considered as a *joint statistic* for overall phrase-level verification. Suppose the detected phrase consists of N subwords, the confidence measure for the phrase is defined as a function of their likelihood ratios.

$$CM = f(LLR_1, \cdots, LLR_N). \tag{3}$$

The phrase is accepted as a valid theory if the corresponding confidence measure exceeds a certain threshold.

We have investigated several functional forms of the confidence measure. The first confidence measure CM_1 is based on frame duration normalization. It is exactly the difference of the two Viterbi scores of the subword models and the corresponding anti-subword models defined as

$$CM_1 = \frac{1}{L} \sum_n (l_n * LLR_n) \tag{4}$$

where l_n is duration of subword n and L is total duration of the phrase, i.e., $L = \Sigma\ l_n$.

The second one CM_2 is based on subword segment-based normalization. It is a simple average of log likelihood ratios of all the subwords.

$$CM_2 = \frac{1}{N} \sum_n LLR_n. \tag{5}$$

The third one CM_3 focuses on less confident subwords rather than averaging all the subwords. This is because some subwords of an incorrect phrase may exactly match the input. For example, the latter part of "November" matches the input "December" and gets good verification scores. In order to reject it, we have to focus on the former parts, which will get poor verification scores. In order to find less confident subwords, we normalize the log likelihood ratio assuming a Gaussian distribution for every subword. The means and variances of log likelihood ratios for all the subwords are estimated with the samples used for training subword and anti-subword models. We denote this normalized log likelihood as LLR_n^*.

$$LLR_n^* = \frac{LLR_n - \mu(LLR_{c(n)})}{\sigma(LLR_{c(n)})} \tag{6}$$

where $\mu(LLR_{c(n)})$ and $\sigma(LLR_{c(n)})$ are the mean and the variance for subword class of n, respectively. Then, we pick up those subwords whose likelihood ratios are less than their means $\mu(LLR_{c(n)})$. Thus, CM_3 is defined as

$$CM_3 = \frac{1}{N} \sum_n \begin{cases} LLR_n^*, & \text{if } LLR_n^* < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The fourth confidence measure CM_4 uses the sigmoid function. This form is used as a loss function for training with the minimum error rate criteria.

$$CM_4 = \frac{1}{N} \sum_n \frac{1}{1 + \exp(-\alpha \cdot LLR_n)}. \tag{8}$$

For every confidence measure, a specific threshold is set up. If its value is below the threshold, the candidate is discarded from the phrase lattice.

V. Sentence Parsing and Verification

A. Sentence Parsing

Parsing algorithms are necessary for combining the verified phrase candidates into sentence hypotheses. We focus on the one-directional left-to-right search. Since trellis parsing

requires much computation with a little improvement of accuracy, we adopt a lattice parsing algorithm [21]. It connects phrase candidates according to their acoustic scores and the semantic constraints. The semantic constraints specify permissible combinations of key-phrase tags. As the acoustic score, we use the score given by the forward-backward search in key-phrase detection.

In order to find the most likely sentence hypothesis efficiently, the stack decoding search is adopted. It extends the best partial hypotheses until a sentence hypothesis is completed. Suppose that the current hypothesis at the top of the stack is $q_0 = \{\boldsymbol{w}_1, \boldsymbol{w}_2\}$, and a new hypothesis is generated by concatenating a phrase $\boldsymbol{w}_3$. The evaluation function for the new hypothesis $q_1 = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$ is computed as an offset from the upper bound score for the whole input, h_0, as follows:

$$\begin{aligned}\hat{f}(\boldsymbol{w}_1\boldsymbol{w}_2\boldsymbol{w}_3) &= h_0 - (h_0 - \hat{f}(\boldsymbol{w}_1)) - (h_0 - \hat{f}(\boldsymbol{w}_2)) \\ &\quad - (h_0 - \hat{f}(\boldsymbol{w}_3)) \\ &= \hat{f}(\boldsymbol{w}_1\boldsymbol{w}_2) - (h_0 - \hat{f}(\boldsymbol{w}_3))\end{aligned}$$

where $\hat{f}(\boldsymbol{w}_i)$ is an evaluation value for a detected phrase $\boldsymbol{w}_i$. The initial hypothesis is $\hat{f}(null) = h_0$. Every time a new phrase is added, its offset is subtracted. The upper bound h_0 is computed in the forward pass of the recognition with the phrase network.

The algorithm is based on the *short-fall method* [27], and the evaluation is A*-admissible. However, its heuristic power is weak in guiding the search efficiently. Especially, in the detection-based parsing which does not assume complete coverage of the whole input, shorter hypotheses with fewer words are likely to be accepted. Thus, we need to modify the algorithm to accommodate the skipped portion. One way to accomplish this is to simply offset a uniform penalty value proportional to the skipped length. This rough approximation assumes noninformative statistic and makes the search suboptimal. To enhance the result, it is desirable that as many filler phrases (including silence) as possible are generated as well as key-phrases in forming sentence hypotheses.

B. Sentence Verification

The sentence verification module makes the final decision on the recognition output. It uses the global acoustic and semantic information on the entire input utterance. While the key-phrase verification makes only local decisions, the sentence verification process combines its results and realizes a similar effect as the conventional utterance verification algorithm, although it attempts to accept the input even if it contains unexpected filler phrases.

The semantic verification process judges if the semantic representation in the output is completed. In dialogue applications, we often observe incomplete utterances; for example, saying a month, "August," without specifying any days of the month. Ideally, they should also be accepted with the assumption that remaining semantic slots will be completed during the subsequent dialogue exchanges. However, unconditional approval of partial sentences invalidates the effect of utterance verification and accepts false alarms as well. Therefore, we reject a sentence hypothesis only if its semantic representation is not completed *and* most of the input segments are rejected by the likelihood ratio tests. When we apply this on the N-best outputs of the sentence hypotheses, the parsing and verification process will continue until a satisfactory one is obtained.

Out-Of-Grammar samples

I will be returning the car on September fourth

August fit August fifth

eight twenty August twentieth

Out-Of-Task samples

uh Decem

Idaho Idaho

(breath only)

Fig. 2. Sample utterances of DATE subtask.

VI. Experimental Evaluation

We have evaluated our algorithms in two spoken dialogue applications; one is "car reservation task" and the other is "movie locator task." The first one involves several interactions of simple utterances, while the latter task is generally completed with a single query of a rather complex sentence. Trials were performed on dialogue systems of the tasks using a speech recognizer. All the data were collected via telephone lines and uttered by general public users.

For evaluation, we define the semantic accuracy in much the same way as the word accuracy. In particular, the semantic error is defined based on the sum of substitution, insertion and deletion errors by matching the content of the semantic slots instead of the recognized words. The semantic accuracy is formulated as follows:

$$\begin{aligned}1 - (&\#\text{ substitution errors } + \#\text{ insertion errors} \\ &+ \#\text{ deletion errors}) / \#\text{ answers.}\end{aligned} \tag{9}$$

This measure demands strict verification, namely to reject extraneous words; otherwise insertion errors are counted.

For an example sentence, "That will be Saturday, December twenty-fourth," the following three semantic slots are the expected output:

[SATURDAY: dy.6] [DECEMBER: mt.12]
[TWENTY FOURTH: dt.24].

For a detailed analysis, the sample utterances are classified into three categories. The classification is purely based on transcription texts and does not reflect acoustic difficulty. *In-grammar* sentences consist of valid phrases and are covered by the conventional sentence grammars. *Out-of-grammar* sentences have out-of-vocabulary or fragmental words, or segments with more than one assignment to a semantic slot. They should be properly interpreted in an ideal dialogue system but are usually not accepted by the rigid sentence

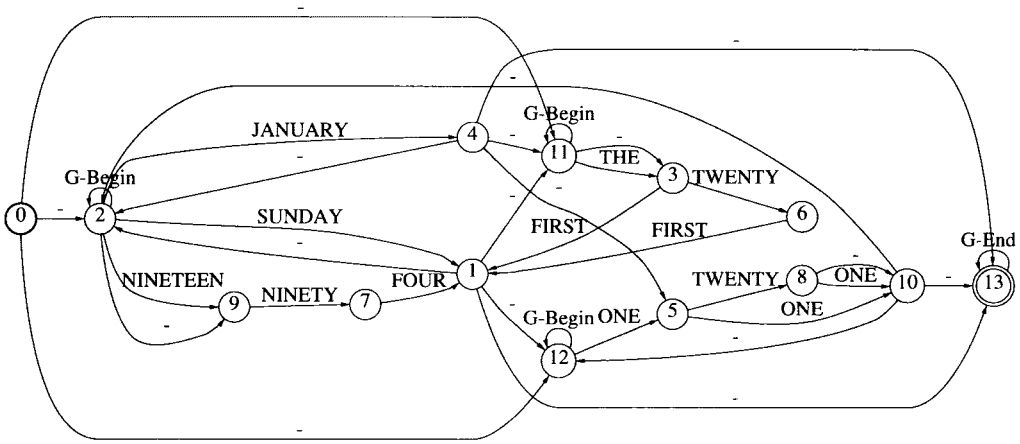

Fig. 3. Phrase network for DATE subtask.

grammars. *Out-of-task* sentences contain no key-phrases and should be rejected. For a unified definition of the semantic accuracy, we prepare a null slot as an answer for them. Thus, the semantic accuracy for out-of-task samples means the correct rejection rate.

A. Car Reservation Task

In the car reservation task, a user is asked to provide all the reservation information by voice so that a rental car reservation form can be completed. The current dialogue management is rigid. The user is prompted to give a reply to a particular request. Specifically, the form fields include the account number, the spelling of the user name, pickup and drop-off locations, dates, times, and the desired car type [1].

We refer to each pair of the prompt and the answer specifying such information as a subtask. For each subtask, a different vocabulary set and a grammar is prepared to improve recognition performance. For example, in the LOCATION subtask, prompting "Please say your pickup location," the system accepts only vocabulary and expressions regarding the rental location.

Here, we choose the DATE subtask for the primary evaluation, because it contains the largest number of samples and typical dialogue phenomena. The total number of samples is 1368. Besides in-grammar utterances, there are 154 out-of-grammar and 91 out-of-task samples. Some examples are shown in Fig. 2.

A simplified phrase network for the DATE subtask is shown in Fig. 3. The phrase subgrammar allows iterations of days of the week, months, days of the month, and years with some constraints. The vocabulary size is 99. In this subtask, no filler phrases are incorporated.

Phrase verification was performed with several confidence measures defined in the previous section. Fig. 4 shows comparison of the confidence measures with the acceptance rate of incorrect phrases (false alarms) versus the false rejection rate of correct phrases. The frame duration-based confidence measure (CM_1) is inferior to the subword segment-based ones. The confidence measure CM_3 that is proposed in this work achieves the best performance. This measure reduces

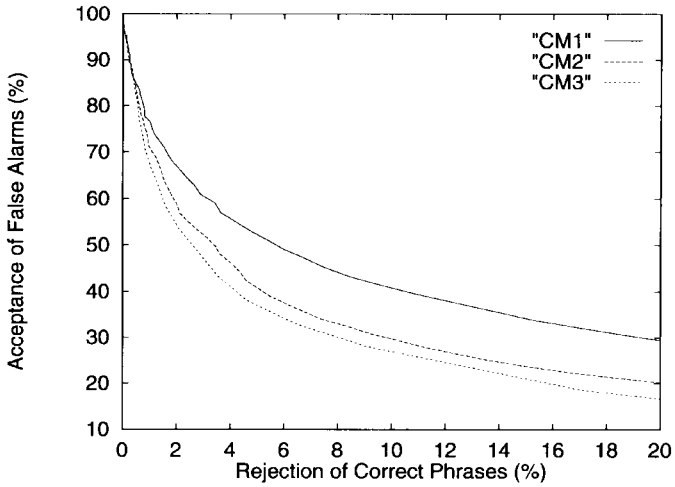

Fig. 4. Effect of phrase verification (DATE subtask).

the false alarms to a half with 2.5% rejection of correct hypotheses.

The fact is also confirmed in Fig. 5 that plots distributions of the confidence measure CM_3 for correct and incorrect hypotheses. There are only a few correct hypotheses in the range of lower half of the distributions for incorrect ones.

This reduction improves the semantic accuracy of out-of-grammar and out-of-task samples as a result of the sentence parsing. Fig. 6 shows the semantic accuracy for each category of samples depending on the threshold values for CM_3. The left-most of the graph corresponds essentially to the baseline detection method without any verification. The best operating point exists between -1.2 and -1.6, below which the accuracy of in-grammar utterances does not change and that of out-of-grammar and out-of-task samples decreases. While the curves for in-grammar and out-of-task utterances are monotonous, there is a performance peak on the out-of-grammar samples that affects an ideal choice of the threshold value.

Then, several confidence measures were compared in semantic accuracy after tuning thresholds for each in Table I. Although the use of any confidence measures improved the accuracy, the frame duration-based confidence measure CM_1 is not as effective as the subword segment-based measures

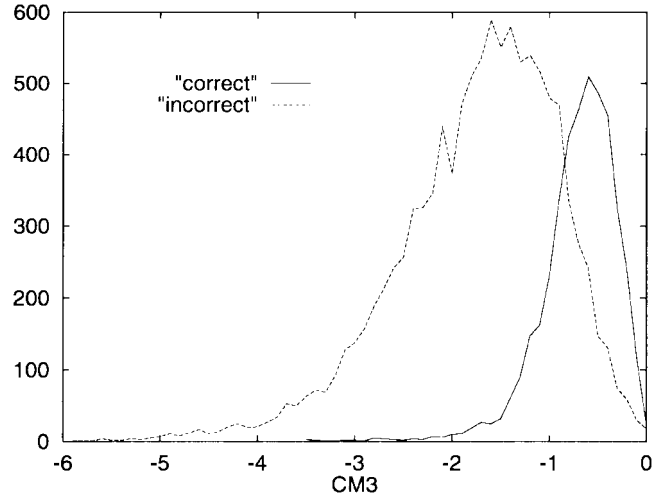

Fig. 5. Distributions of CM3 (DATE subtask).

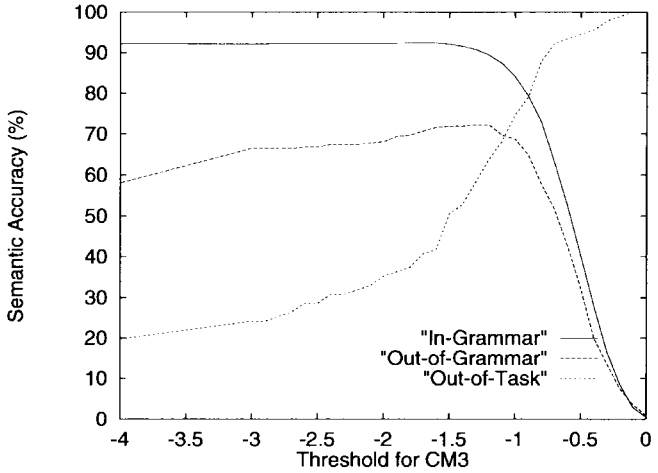

Fig. 6. Semantic accuracy versus threshold (DATE subtask).

($CM_2 \sim CM_4$), as observed in the phrase verification performance. The result matches with the previous work [28]. The proposed confidence measure CM_3 that focuses on less confident subwords leads to the best performance.

Next, several approaches for speech understanding were investigated. Here, sentence verification was incorporated. For comparison, a rigid grammar was also applied. It is fundamentally the same as the one used for the field trial, and uses the constraint of typical sequences of phrases, which detection does not assume. We also compared with the decoding followed by the verification procedure as in [28]. For phrase verification, CM_3 was adopted. The same beam width was used for all methods.

The results are listed in Table II. Strictly speaking, both detection with a rigid grammar and decoding with a phrase network (relaxed grammar) are possible. However, decoding with the phrase network is unfairly compared because it does not involve interphrase constraints and optimizations. And detection with a rigid grammar makes no difference in performance from decoding, because it just delimits the sentence into pieces and assembles them with the same constraints. Therefore, we simply refer to **decoding** as decoding with a rigid grammar and **detection** as detection with a phrase network.

It is clear that our detection strategy outperforms the conventional decoding scheme. It achieves much higher accuracy for out-of-grammar samples while keeping comparable performance for in-grammar ones. Detection with the phrase network almost doubles the accuracy for out-of-grammar samples, and the use of phrase verification improves it further. The verification applied after decoding improves the rejection performance for out-of-task utterances, but it is not so effective in recognizing out-of-grammar samples. This is because key-phrases cannot be recovered from the result of the initial decoding processing with the rigid grammar. The sentence-level verification has little effect, but it improves rejection of out-of-task utterances.

We have also done experiments on other subtasks in the car reservation task. The results on TIME, and LOCATION subtasks are shown in Tables III and IV, respectively. In these results, the confidence measure CM_3 was used for phrase rejection, although there was little difference observed among the choices of CM_2, CM_3, and CM_4. In all these subtasks, much the same tendency is confirmed as in the DATE subtask. The detection-based strategy outperforms the decoding methods, and the phrase verification improves the accuracy for out-of-grammar utterances.

TABLE I
SEMANTIC ACCURACY WITH SEVERAL CONFIDENCE MEASURES (DATE SUBTASK)

	in-grammar samples	out-of-grammar samples	out-of-task samples	total
number of samples	1123	154	91	1368
no verification	92.2%	58.1%	19.8%	86.2%
$CM1$	91.7%	68.1%	34.1%	87.3%
$CM2$	91.9%	71.3%	41.8%	88.0%
$CM3$	92.3%	71.6%	41.8%	88.5%
$CM4$	91.8%	71.0%	44.0%	88.0%

B. Movie Locator Task

The movie locator task allows a user to make an inquiry on movies being played at theaters. The field trials were designed to deal with information in the Chicago area. In the specification, a user can ask about movie titles, theaters, or the time, by specifying a movie title, a movie category, a theater, or a location area. Typically, the session completes in a single utterance of the query followed by a system reply. But the utterances are full or complex sentences and involve multiple phrases as well as extraneous words. We observed a variety of out-of-grammar samples, which constitute more than 25% of the collected samples.

Examples of these utterances are listed in Fig. 7. In the third example of the out-of-grammar samples, the specification like "between eight and ten" is not allowed in the task. The fourth example contains the movie title *Citizen Kane*, which was no longer played at the theaters. As in the fifth example, it is more likely that a sentence contains at least one appropriate semantic

TABLE II
SEMANTIC ACCURACY WITH SEVERAL APPROACHES (DATE SUBTASK)

	in-grammar samples	out-of-grammar samples	out-of-task samples	total
number of samples	1123	154	91	1368
decoding (with rigid grammar)	92.7%	29.4%	18.7%	83.4%
+ phrase verification ($CM3$)	92.8%	42.3%	39.6%	85.6%
+ sentence verification	92.8%	41.3%	48.4%	85.7%
detection (with phrase network)	92.2%	58.1%	19.8%	86.2%
+ phrase verification ($CM3$)	92.3%	71.6%	41.8%	88.5%
+ sentence verification	92.2%	71.6%	51.6%	88.7%

In-Grammar Samples

Where is Forrest Gump playing near Wheaton ?

What-s playing at the Arcada theater in Saint Charles ?

What time is the Specialist playing at the Stratford square theaters ?

Out-Of-Grammar Samples

What fa fam what family movies are playing in Evanston ?

Can you tell me if there are any comedies playing at Stratford Mall theater ?

Where is Forrest Gump playing near Wheaton between eight and ten ?

Where is Citizen Kane playing near Naperville ?

What is the movie Nell about ?

Out-Of-Task Samples

What does rated R mean ?

What does the movie cost ?

Fig. 7. Sample utterances of MOVIE task.

TABLE III
SEMANTIC ACCURACY (TIME SUBTASK)

	in-grammar samples	out-of-grammar samples	out-of-task samples	total
number of samples	818	110	63	991
decoding	87.7%	11.1%	27.0%	79.1%
+ phrase verification	86.9%	26.3%	58.7%	80.7%
+ sentence verification	86.9%	24.7%	60.3%	80.6%
detection	86.6%	46.3%	20.6%	81.1%
+ phrase verification	85.6%	64.7%	55.6%	82.9%
+ sentence verification	85.6%	62.6%	58.7%	82.8%

TABLE IV
SEMANTIC ACCURACY (LOCATION SUBTASK)

	in-grammar samples	out-of-grammar samples	out-of-task samples	total
number of samples	681	99	131	911
decoding	94.2%	16.1%	26.0%	79.0%
+ phrase verification	93.3%	21.9%	41.2%	80.4%
+ sentence verification	93.3%	21.9%	41.2%	80.4%
detection	92.6%	40.1%	20.6%	79.7%
+ phrase verification	91.2%	59.1%	35.1%	82.1%
+ sentence verification	91.1%	57.7%	37.4%	82.1%

slot (movie title *Nell* in this example), even if the query itself is not relevant. As a result, the ratio of the out-of-task samples is smaller.

The number of utterances used for evaluation is 2303, and the vocabulary size is 474.

The phrase network was derived by connecting parallel phrase subgrammars, instead of using a rigid grammar to cover whole sentences. We first describe phrase subgrammars for every semantic slot. The phrase subgrammars were semi-automatically derived by connecting keywords and adjacent functional words. Then, the phrase grammar automata were connected with some constraints in a recurrent way using the preceded trial data. Sentence verification was not tested for this task because there were only a few out-of-task utterances in the test database.

The sentence understanding results are listed in Table V. Much the same tendency as in the car reservation task is confirmed. The detection strategy achieves higher accuracy

TABLE V
SEMANTIC ACCURACY (MOVIE-2 TASK)

	in-grammar samples	out-of-grammar samples	out-of-task samples	total
number of samples	1662	601	40	2303
decoding	78.1%	33.5%	5.0%	65.6%
+ verification ($CM3$)	76.8%	42.4%	30.0%	67.3%
detection	79.2%	44.8%	5.0%	69.5%
+ verification ($CM1$)	78.9%	45.4%	7.5%	69.4%
+ verification ($CM2$)	79.5%	47.4%	17.5%	70.4%
+ verification ($CM3$)	78.0%	51.3%	30.0%	70.5%
+ verification ($CM4$)	78.5%	49.1%	27.5%	70.2%

than the decoding one, and the verification process improves further. Among the confidence measures, CM_1 is worse than the others.

VII. DISCUSSIONS AND CONCLUSION

We have proposed a key-phrase detection and verification approach oriented for flexible spoken language systems. The combination of key-phrases realizes wider coverage than conventional sentence grammars. The combined detection and verification strategy focuses on the key-phrases and suppresses the false alarms in the out-of-vocabulary or out-of-grammar portions.

The constraint of the phrase network and the verification procedure significantly improves the detection rate. We have also studied confidence measures for phrase candidates based on a subword-based verifier, and proposed a new measure sensitive to incorrectly recognized subwords. With this measure, the false alarms are reduced to a half with a slight false rejection. Since the key-phrases are tagged with semantic slots, their detection directly leads to robust understanding. Sentence parsing and verification are performed using this information. They are effective for rejecting out-of-task utterances especially in limited task domains.

The experimental results on several tasks demonstrate that the proposed approach is more effective than the conventional decoding with rigid grammars. It drastically improves the accuracy for out-of-grammar utterances while keeping comparable performance for in-grammar ones. The verification applied after decoding is effective only for rejecting out-of-task utterances but does not realize flexible understanding of out-of-grammar ones.

The key properties of our framework are portability and generality. Both the detection and verification are vocabulary-independent subword-based, thus applicable to a variety of new tasks. Moreover, the language model of the key-phrase network is easily derived from task specifications. Our ongoing research includes refinement of filler phrases using other large corpora. It is one approach of task adaptation of language model without assuming task-specific data, and will complement our framework [29].

The integration of the proposed speech understanding strategy with a dialogue manager is an important issue for further studies to complete a flexible spoken dialogue system. The confidence measures obtained in the verification process will be useful in the user interface design to decide when and how to confirm users' answers, when and what kind of voice repair is needed. We believe our framework of combined detection and verification will contribute toward designing intelligent speech interfaces.

ACKNOWLEDGMENT

The authors wish to thank D. Brown of AT&T Labs and J. Wisowaty of Bell Labs for providing the task and grammar specifications and sharing the test data for the car reservation and the movie locator tasks. The authors also thank W. Chou of Bell Labs for many helpful discussions, and Prof. Doshita of Kyoto University and Dr. Atal of Bell Labs for this collaborative research opportunity.

REFERENCES

[1] S. M. Marcus, D. W. Brown, and R. G. Goldberg, "Prompt constrained natural language—Evolving the next generation of telephony services," in *Proc. ICSLP*, 1996, pp. 857–860.
[2] D. A. Dahl, "Expanding the scope of the ATIS task: The ATIS-3 corpus," in *Proc. ARPA Human Language Technology Workshop*, 1994, pp. 43–48.
[3] R. Pieraccini *et al.*, "A speech understanding system based on statistical representation of semantics," in *Proc. IEEE-ICASSP*, vol. 1, pp. 193–196, 1992.
[4] S. Miller, R. Schwartz, R. Bobrow, and R. Ingria, "Statistical language processing using hidden understanding models," in *Proc. ARPA Human Language Technology Workshop*, 1994, pp. 278–282.
[5] R. C. Rose, "Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition," *Comput. Speech Lang.*, vol. 9, pp. 309–333, 1995.
[6] H. Tsuboi and Y. Takebayashi, "A real-time task-oriented speech understanding system using keyword-spotting," in *Proc. IEEE-ICASSP*, vol. 1, pp. 197–200, 1992.
[7] J. R. Rohlicek, *et al.*, "Phonetic training and language modeling for word spotting," in *Proc. IEEE-ICASSP*, vol. 2, pp. 459–462, 1993.
[8] M. Weintraub, "Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system," in *Proc. IEEE-ICASSP*, vol. 2, pp. 463–466, 1993.
[9] J. G. Wilpon *et al.*, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1870–1878, 1990.
[10] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. IEEE-ICASSP*, 1990, pp. 129–132.
[11] W. Ward, "Understanding spontaneous speech: The PHOENIX system," in *Proc. IEEE-ICASSP*, 1991, pp. 365–367.
[12] E. Jackson *et al.*, "A template matcher for robust NL interpretation," in *Proc. DARPA Speech Natural Lang. Workshop*, 1991, pp. 190–194.
[13] S. Seneff, "Robust parsing for spoken language systems," in *Proc. IEEE-ICASSP*, 1992, vol. 1, pp. 189–192.
[14] D. Stallard and R. Bobrow, "Fragment processing in the DELPHI system," in *Proc. DARPA Speech and Natural Language Workshop*, 1992, pp. 305–310.
[15] B. Suhm and A. Waibel, "Toward better language models for spontaneous speech," in *Proc. ICSLP*, 1994, pp. 831–834.
[16] E. P. Giachin, "Phrase bigrams for continuous speech recognition," in *Proc. IEEE-ICASSP*, 1995, pp. 225–228.
[17] S. Deligne and F. Bimbot, "Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams," in *Proc. IEEE-ICASSP*, 1995, pp. 169–172.
[18] R. C. Rose, B.-H. Juang, and C.-H. Lee, "A training procedure for verifying string hypotheses in continuous speech recognition," in *Proc. IEEE-ICASSP*, 1995, pp. 281–284.
[19] R. A. Sukkar *et al.*, "Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training," in *Proc. IEEE-ICASSP*, 1996, pp. 518–521.
[20] M. Rahim *et al.*, "Discriminative utterance verification using minimum string verification error (MSVE) training," in *Proc. IEEE-ICASSP*, 1996, pp. 3585–3588.
[21] T. Kawahara, N. Kitaoka, and S. Doshita, "Concept-based phrase spotting approach for spontaneous speech understanding," in *Proc. IEEE-ICASSP*, 1996, pp. 291–294.
[22] C.-H. Lee *et al.*, "Improved acoustic modeling for large vocabulary continuous speech recognition," *Comput. Speech Lang.*, vol. 6, pp.

103–127, 1992.
[23] W. Chou, *et al.*, "An algorithm of high resolution and efficient multiple string hypothesization for continuous speech recognition using inter-word models," in *Proc. IEEE-ICASSP*, vol. 2, pp. 153–156, 1994.
[24] C.-H. Lee *et al.*, "A study on task-independent subword selection and modeling for speech recognition," in *Proc. ICSLP*, 1996, pp. 1816–1819.
[25] W. Chou, C.-H. Lee, and B.-H. Juang, "Minimum error rate training based on the N-best string models," in *Proc. IEEE-ICASSP*, 1993, vol. 2, pp. 652–655.
[26] R. A. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 420–429, 1996.
[27] W. A. Woods, "Optimal search strategies for speech understanding control," *Artif. Intell.*, vol. 18, pp. 295–326, 1982.
[28] E. Lleida and R. C. Rose, "Efficient decoding and training procedures for utterance verification in continuous speech recognition," in *Proc. IEEE-ICASSP*, 1996, pp. 507–510.
[29] T. Kawahara, S. Doshita, and C.-H. Lee, "Phrase language models for detection and verification-based speech understanding," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 49–56.

Tatsuya Kawahara (M'91) received the B.E. degree in 1987, the M.E. degree in 1989, and the Ph.D. degree in 1995, from Kyoto University, Kyoto, Japan, all in information science.

In 1990, he became a Research Associate at the Department of Information Science, Kyoto University. From 1995 to 1996, he was a visiting researcher at Bell Laboratories, Murray Hill, NJ. Currently, he is an Associate Professor at the School of Informatics, Kyoto University. He has been working on speech recognition and understanding. His current interest includes key-phrase detection, language modeling, and search algorithms. He is also engaged in projects on a Japanese speech dictation system and spoken dialogue systems.

Chin-Hui Lee (S'79–M'81–SM'90–F'97) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree from Yale University, New Haven, CT, in 1977, and the Ph.D. degree from University of Washington, Seattle, in 1981, all in electrical engineering.

In 1981, he joined Verbex Corporation, Bedford, MA, and was involved in research work on connected word recognition." In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, CA, where he engaged in research in speech coding, speech recognition and signal processing for the development of the DSC-2000 Voice Server. Since 1986, he has been with Bell Laboratories, Murray Hill, NJ, where he is now a Distinguished Member of Technical Staff and Head of Dialogue Systems Research Department at Bell Labs, Lucent Technologies. His current research interests include signal processing, speech modeling, adaptive and discriminative modeling, speech recognition, speaker recognition and spoken dialogue processing. He has published more than 170 papers in journals and international conferences and workshops on the topics in automatic speech and speaker recognition. His research scope is reflected in his edited book *Automatic Speech and Speaker Recognition: Advanced Topics* (Boston, MA: Kluwer, 1996).

Dr. Lee was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING (1991 to 1995). He was a member of the ARPA Spoken Language Coordination Committee during the same period. Since 1995, he has also been a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS), in which he serves as Chairman. In 1996 he helped promote the newly formed SPS Multimedia Signal Processing (MMSP) Technical Committee and is now a member. He vis a recipient of the 1994 SPS Senior Award and the 1997 SPS Best Paper Award in Speech Processing. He was a winner of the prestigious Bell Laboratories President Gold Award in 1997 for his contributions to the Bell Labs Automatic Speech Recognition algorithms and products.

Biing-Hwang Juang (S'79–M'81–SM'87–F'92) received the B.Sc. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 1979 and 1981, respectively.

In 1978, he did research on vocal tract modeling at Speech Communications Research Laboratory (SCRL). He then joined Signal Technology, Inc., in 1979 as a Research Scientist, working on signal and speech related topics. Since 1982, he has been with Bell Laboratories, Murray Hill, NJ, where he is engaged in a wide range of communication related research activities, from speech coding and speech recognition to multimedia communications. He is currently Head of the Acoustics and Speech Research Department in the Multimedia Communications Research Laboratory. He has published extensively and holds a number of patents in the area of speech communication and communication services. He is a co-author of the *Fundamentals of Speech Recognition* (Englewood Cliffs, NJ: Prentice-Hall, 1993).

Dr. Juang received the 1993 Senior Paper Award, the 1994 Senior Paper Award, and the 1994 Best Signal Processing Magazine Paper Award, all from the IEEE Signal Processing Society. He was an Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (1986–1988), the IEEE TRANSACTIONS ON NEURAL NETWORKS (1992–1993), and the *Journal of Speech Communication* (1992–1994). He has served on the Digital Signal Processing and the Speech Technical Committees as well as the Conference Board of the IEEE Signal Processing Society, and was Chairman of the Technical Committee on Neural Networks for Signal Processing (1991–1993). He is currently Editor-in-Chief of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He also serves on international advisory boards.