KEY-PHRASE DETECTION AND VERIFICATION FOR FLEXIBLE SPEECH UNDERSTANDING

*Tatsuya Kawahara** *Chin-Hui Lee* *Biing-Hwang Juang*

Bell Laboratories, Murray Hill, NJ 07974-0636

ABSTRACT

A novel framework of robust speech understanding is presented. It is based on a detection and verification strategy. It extracts the semantically significant parts and rejects the irrelevant parts rather than decoding the whole utterances. There are two key features in our strategy. Firstly, discriminative verifier is integrated to suppress false alarms. It uses anti-subword models specifically trained to verify the recognition results. The second feature is the use of a keyphrase network as the detection unit. It embeds stochastic constraint of keyword and keyphrase connections to improve the coverage and detection rates. The automatic generation of the keyphrase network structure is also addressed. This top-down variable-length language model can be trained with a small corpus and ported to different tasks. This property coupled with the vocabulary-independent detector and verifier enhances the portability of our framework.

1. INTRODUCTION

As spoken language systems or spoken dialogue systems are being evaluated for wider usage in real-world applications, it is found that they are not sufficient to cope with the utterance variation inherent in a large user population. The conventional systems try to decode the whole utterance with the fixed vocabulary and language models and match every part of the input with some words uniformly. The use of a rigid grammar is effective for typical in-grammar sentence patterns. However, in real-world environments, we have observed a large number of out-of-grammar utterances even after the task grammars had been tuned by human experts during the trial period. These samples include extraneous words, hesitations, repairs and unexpected expressions.

On the other hand, most of the mis-recognized utterances contain some keyphrases that are task-related and may lead to partial or full understanding. Flexible speech understanding should be able to detect semantically significant parts and reject irrelevant parts. In a specific task domain of a transaction or information retrieval system, it is possible to make sense with keywords or keyphrases. Therefore, the approach based on their detection is attractive. By relaxing the grammatical constraints and focusing on the keywords and keyphrases, it will accept a wider variety of utterances than sentence grammars can.

*This work has been carried out while visiting from Dept. of Information Science, Kyoto University, Kyoto 606-01, Japan

Especially in dialogue-based systems, it is possible to set up grammars according to the dialogue state and apply to a large-scale task. Combined with a flexible dialogue manager, the detection-based approach realizes partial understanding and disambiguation of unclear parts through the following dialogue session.

In this paper, we propose a keyphrase detection approach that realizes flexible understanding. Based on a review of the conventional word spotting approaches, which often suffers from excessive false alarms and is hard to apply to a variety of tasks, our framework features the discriminative verification and the keyphrase network structure to improve the detection performance and the system portability.

2. SYSTEM OVERVIEW

The outline of our detection-based speech understanding system consists of the following steps.

1. keyphrase detection

   A set of keyphrases are detected using a set of phrase sub-grammars specific to the dialogue state. The keyphrases are labeled with semantic tags, which are useful in the sentence-level parsing.

2. keyphrase verification

   The detected keyphrases are verified and assigned confidence measures. The process eliminates false alarms and rescores the verified candidates. The verifier is constructed with *anti-subword models* which test the individual subwords of the recognized results.

3. sentence parsing

   The keyphrase candidates are connected into sentence hypotheses using the task-specific semantic knowledge sources. A stack decoder is used to search the optimal hypotheses that satisfy the semantic constraints[1].

4. sentence rescoring

   The semantically valid sentence hypotheses are verified and rescored with detailed classification and verification models by reprocessing the speech input.

Step 1 and 2 are performed simultaneously with the respective modules. Detail of these steps are discussed in the following sections, as they are novel parts of our framework.

3. KEYPHRASE DETECTION AND VERIFICATION

The conventional decoding scheme cannot handle out-of-vocabulary words and out-of-grammar utterances, because it assumes the uniform constraint on the whole utterance.

As a more robust strategy, the word spotting scheme[2] has been studied. They are classified into two approaches with respect to the modeling of non-keyword parts.

One is the application of a large vocabulary speech recognition[3][4]. It incorporates as much lexical knowledge as possible. However, it cannot model the ill-formed phenomena such as self-repairs and hesitations, which are occasionally found in spontaneous speech in the real environment. It does not solve the problem inherent to the uniform decoding scheme. And the large vocabulary spontaneous speech recognition is not a realistic solution both in performance and efficiency, especially when the task domain is limited.

The other is simple word spotting with a simple garbage model or the parallel network of subword models[5][6]. They are not sufficient models of non-keywords and the keyword models are easily matched with irrelevant parts, which causes so many false alarms that cannot be handled with the subsequent processing. Most of the work tune the keyword models and the garbage model in vocabulary-dependent manners, sacrificing the advantage of subword-based recognition. Thus, the approach can be applied only to very small vocabulary tasks such as digit recognition.

3.1. Detection & Verification Strategy

Our goal is to realize ideal detection mechanism that neither uses large vocabulary knowledge, nor matches irrelevant parts, in a general framework of subword-based recognition. One of the most significant problem is that conventional recognizers do not know how confident its outputs are. Therefore, we have studied verification methods to perform hypothesis test on the recognized result[7]. In this work, we integrate the verification technique in order to realize reliable and robust detection.

Another phenomena in spontaneous speech to support this strategy is keyphrases are more clearly uttered than ill-formed parts, and thus expected to be easily verified.

3.2. Keyphrase Network as Detection Unit

The other feature in our strategy is to use a keyphrase network as the detection unit instead of keywords.

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.

A keyphrase consists of one or a few keywords and functional words, for example, 'on Sunday' or 'ninety nineteen six'. Even in spontaneous speech, they are uttered without break.

Moreover, we incorporate the constraint that inhibits the impossible connection of keyphrases. As a whole, the detection unit is a network of keyphrase sub-grammar automata with their permissible connection or iteration. We call this a keyphrase network. It is easily extended to a stochastic language model by estimating the connection weight. The constraint achieves wider coverage with modest perplexity than sentence-level grammars.

The keyphrases are tagged with conceptual information. In fact, we define our keyphrases so as to correspond with semantic slots such as date and place. Unlike bottom-up phrases defined by N-gram scheme, our top-down phrases are directly mapped into semantic representations. Thus, detection of them directly leads to robust understanding.

We also define filler phrases that often accompany the keyphrases. The automatic definition of keyphrases and filler phrases are discussed in Section 5.

3.3. Detection Algorithm

The detection algorithm adopted in this work is based on the forward-backward two-pass search[8], although a one-pass detection is possible in theory. The major modification dedicated to the detection purpose is the use of hypothesis merging and pruning.

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

It is implemented by marking merging states of the keyphrase network. A merging state corresponds to the node where the keyphrases or filler phrases are completed and further extension starts the next new phrases.

When a hypothesis popped by the stack-decoder is tagged as a complete phrase to be output, we extend one more word and align the phrase with the best extension. If this node is 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 and sub-optimally produces the correct N-best keyphrase candidates by the order of their scores. It terminates at a desired number of phrases or a certain score threshold.

3.4. Verification Algorithm

The verification of the detected phrases is performed based on the subword-level test[7]. For every subword model, a corresponding anti-subword model is trained specifically for the verification task. The anti-subword model is constructed by clustering the highly confusing subword classes.

For every subword of the phrase, a verification score is computed, as a result of likelihood ratio test, by offsetting the likelihood of its anti-subword model. By combining the subword-level verification scores, the verification and rescoring of the phrase candidate are done.

Phrase verification focuses on less confident subwords, because some subwords of a false phrase can exactly match the input. Concretely, we compute the normalized sum of the likelihoods for the subwords that get negative as a result of subtracting those of the anti-subword models. If this phrase verification score is below a certain threshold, the candidate is discarded from the phrase lattice.

For rescoring the verified candidates, all the subwords are considered to compute a confidence measure.

In this way, vocabulary independent verification is realized. The use of anti-subword model as a reference is more discriminative than unconstrained decoding of subword models, because it can focus on the critical subwords of the phrase and the anti-subword models are dedicated to the verification of the specific subword. Moreover, it has the ability to reject substitution errors by the recognizer.

4. PRELIMINARY EXPERIMENTAL RESULTS

The algorithms have been implemented and evaluated on several sub-tasks intended for spoken dialogue systems. All the data are collected via telephone lines and uttered by the general public.

The active vocabulary size of the systems ranges from 100 to 400, depending on the grammar used for the specific dialogue state. It is observed that 20∼30% of the utterances are out-of-grammar, even after the grammars are tuned by a human.

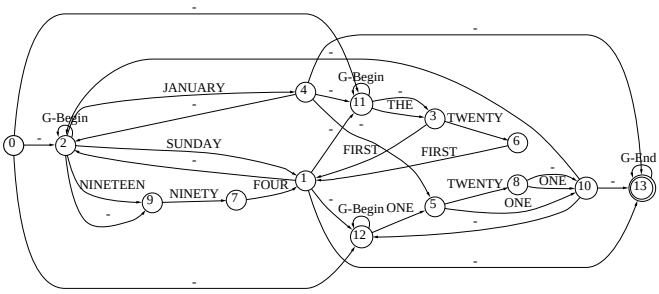

Figure 1. Keyphrase network for date phrases

```
input: I WILL BE RETURNING ON SEPTEMBER FOURTH

 t=277-330 f=13977. FOURTH/dt.4
 t=216-276 f=13977. SEPTEMBER/mt.9
 t=169-215 f=13977. APRIL/mt.4  ---> reject
 t=126-168 f=13977. JUNE/mt.6   ---> reject
 t= 75-125 f=13977. NOVEMBER/mt.11 ---> reject
 t=187-217 f=13962. ONE/dt.1
 t=160-186 f=13961. SIX/mt.6
 t=126-159 f=13962. THREE/dt.3
 t=126-164 f=13954. THREE/dt.3  ---> reject
 t=113-165 f=13940. SATURDAY/dy.6  ---> reject
 t= 71-112 f=13943. FRIDAY/dy.5  ---> reject

output of parsing remaining phrases:
[ [SEPTEMBER/mt.9 216-276] [FOURTH/dt.4 277-330] ].
```

Figure 2. Example of detection and verification process

The simplified network structure of keyphrase network used for the date expressions is shown in Figure 1.

An example of the detection and verification process using the above network is illustrated in Figure 2. It lists the the detected phrases that are semantically tagged such as 'dt.4 (date 4)', as well as the time alignments (t) and the detection scores (f). Some of them are rejected by the verification process, and the rest are parsed with the semantic constraints of the sub-task. The procedure to rescore the verified candidates has not been implemented yet.

Table 1. Semantic accuracy for date expressions

	in-grammar	out-of-grammar	total
decoding (sent. grammar)	92.7%	35.4%	86.6%
decoding (phrase net.)	89.9%	23.1%	82.8%
detection	92.2%	54.1%	88.1%
detection+verification	92.1%	59.9%	88.6%

The search for the most likely sentence hypotheses that satisfy the semantic constraints is realized by a stack decoder[1]. Here, we use the simplest semantic constraint that only one value can be assinged to each semantic slot. The semantic accuracy is defined in much the same way as the word accuracy. It is based on the sum of substitution, insertion and deletion errors by matching the content of the semantic slots instead of recognized words. This measure demands the strict verification, namely to reject extraneous words. Otherwise insertion errors are counted.

Table 1 shows results on 1277 samples of the date expressions. In total, they have 2754 semantic slots to be recognized such as month and day of month. The semantic accuracy is computed for in-grammar and out-of-grammar samples separately, too. The out-of-grammar samples are defined as those that have out-of-vocabulary or fragmental words, or more than one assignments to a semantic slot.

The first row is the result of decoding with a rigid sentence grammar. The rests use the phrase network. The results of decoding, detection-only and detection combined with verification are listed in this order. The use of a rigid grammar gets the best performance for in-grammar samples. However, the decoding scheme hardly copes with out-of-grammar sentences. The detection and verification strategy gets the best accuracy for out-of-grammar utterances.

5. AUTOMATIC DEFINITION OF KEYPHRASES

In order to make our approach applicable to different tasks easily, we address the problem of automatic definition of the keyphrases. It is also formulated as learning of a variable-length language model. While the conventional N-gram scheme[9][10][11] is completely bottom up and searches for all possible combinations of words, our definition of the concept-based phrase is top-down.

It starts with keywords w_0 defined with the task specification as the core of the semantic slots such as weekdays and city names. As a preprocessing, these keywords are merged and substituted into abstract non-terminal symbols, for example, WEEKDAY for Sunday to Saturday, CITY for Atlanta, Philadelphia, etc. Then adjacent 'sticky' words w are concatenated to achieve wider coverage and smaller perplexity. Concretely, concatenation is made if both of the following two measures exceed thresholds:

$$\text{stickiness} = p(w, w_0)/p(w)\cdot p(w_0) = p(w|w_0)/p(w)$$

$$\text{coverage} = p(w, w_0)$$

This procedure is iterated to grow in both forward and backward directions to longer phrases. It terminates when the other core is encountered or no more words can be added. The algorithm works with a reasonable size of corpus, because it focuses on frequent patterns and abstract structures. The obtained phrases are tagged with semantic attributes associated with the keywords. The core can be a combination of several words. To obtain filler phrases, we pick up other core words that are not among the set of keywords but occur frequently in the corpus.

The set of phrases are transformed into a phrase network. A skipping transition is added if concatenation of a word to the phrase decreases the coverage. Initial and final nodes are also automatically determined. Minimization of the network is performed to improve computation efficiency and the coverage. At present, this process is done manually.

The statistical information used for determining phrases can be embedded in the network as a network transition weight in the same way as the variable-length N-gram.

Figure 3 shows typical examples of the phrases generated from the sentence sets used in the recognition experiment. The symbols of capital letters represent non-terminals for keywords. N is for cardinal numbers and Nth is for ordinal numbers. It is confirmed that the set of phrases brings a slightly more constrained network than Figure 1.

Even task-independent knowledge such as general phrases can be extracted. This property is significant in deploying the system in a new task domain and improving the portability of the language model. For example, the language model for the date expressions trained with a rich corpus can be used in some other task that needs the same query. A subgrammar also can be a part of a larger network that includes the semantic tag.

Figure 4 shows some examples of the phrases obtained from ATIS-I corpus of 13099 sentences. The frequent patterns of date expressions are successfully extracted, although some mismatch with Figure 3 is observed due to the difference of the prompt to the users. Figure 4 also lists some carrier phrases which were used for the queries and might be useful in different tasks.

```
MONTH Nth nineteen N2 N1,  WEEKDAY MONTH Nth,
N N N2 N1,  MONTH N nineteen N2 N1, Nth of MONTH
MONTH Nth,  the Nth,  MONTH N,  MONTH the Nth,  N N
```

Figure 3. Sample of date phrases generated from the corpus upon the prompt "On what date ... ?"

```
[date phrase]
on MONTH Nth nineteen N2 N1,  WEEKDAY MONTH Nth,
next WEEKDAY,  on WEEKDAY morning,  on a WEEKDAY
Nth of MONTH

[filler/carrier phrase]
i'd like to make a reservation
show me all the flights from
can you tell me the
```

Figure 4. Sample of phrases generated from ATIS corpus

6. SUMMARY

We have proposed a combined detection and verification strategy to realize flexible speech understanding. It extracts useful keyphrases from spontaneous speech while suppressing the false alarms. Although we need further study on the rejection and rescoring methods, the preliminary experiment shows that the proposed strategy works effectively.

We have also adopted the keyphrase network as the language model. The constraint is also effective in improving the detection accuracy. The keyphrases are tagged with semantic attributes and their detection directly leads to robust understanding. The automatic generation method of the keyphrase structure is also presented. It is confirmed that the generated structure is useful. We are studying whether a task-independent statistical module of the phrase structures can be extracted and utilized.

Portability and generality are significant property of our framework. Both the detection and verification are vocabulary independent subword-based, thus applicable to a variety of large-vocabulary tasks. Moreover, the language model of the keyphrase network can be ported to different tasks and trained with a small corpus.

REFERENCES

[1] T.Kawahara, N.Kitaoka, and S.Doshita. Concept-based phrase spotting approach for spontaneous speech understanding. In *Proc. ICASSP*, 1996.

[2] R.C.Rose. Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition. *Computer Speech and Language*, 9(9):309–333, 1995.

[3] J.R.Rohlicek, P.Jeanrenaud, K.Ng, H.Gish, B.Musicus, and M.Siu. Phonetic training and language modeling for word spotting. In *Proc. ICASSP*, volume 2, pages 459–462, 1993.

[4] M.Weintraub. Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system. In *Proc. ICASSP*, volume 2, pages 463–466, 1993.

[5] J.G.Wilpon, L.R.Rabiner, C.H.Lee, and E.R.Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. Acoust., Speech & Signal Process.*, 38(11):1870–1878, 1990.

[6] R.C.Rose and D.B.Paul. A hidden Markov model based keyword recognition system. In *Proc. ICASSP*, pages 129–132, 1990.

[7] R.A.Sukkar C.-H.Lee. Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition. *IEEE Trans. Speech & Audio Process.*, 4(to appear), 1996.

[8] W.Chou, T.Matsuoka, B.-H.Juang, and C.-H.Lee. An algorithm of high resolution and efficient multiple string hypothesization for continuous speech recognition using inter-word models. In *Proc. ICASSP*, volume 2, pages 153–156, 1994.

[9] B.Suhm and A.Waibel. Towards better language models for spontaneous speech. In *Proc. ICSLP*, pages 831–834, 1994.

[10] E.P.Giachin. Phrase bigrams for continuous speech recognition. In *Proc. ICASSP*, pages 225–228, 1995.

[11] S.Deligne and F.Bimbot. Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams. In *Proc. ICASSP*, pages 169–172, 1995.