6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

ERROR RECOVERY AND SENTENCE VERIFICATION USING STATISTICAL PARTIAL PATTERN TREE FOR CONVERSATIONAL SPEECH

Chung-Hsien Wu, Yeou-Jiunn Chen, and Cher-Yao Yang

Department of Computer Science and Information Engineering,
Cheng Kung University, Tainan, Taiwan.

10.21437/ICSLP.2000-889

ABSTRACT

In this paper, in order to deal with the problems of disfluencies in conversational speech, a partial pattern tree (PPT) and a PPT-based statistical language model are proposed. A partial pattern is defined to represent a sub-sentence with a key-phrase and some optional/ functional phrases. The PPT is an integrated tree structure of the partial patterns generated from the training sentences and used to model the n-gram and grammatical constraints. In addition, a PPT merging algorithm is also proposed to reduce the number of partial patterns with similar syntactic structure by minimizing an objective cost function. Using the PPT, the undetected/misdetected errors due to disfluencies can be recovered. Finally, a sentence verification approach is proposed to re-rank the recovered sentences generated from the PPT. In order to assess the performance, a faculty name inquiry system with 2583 names has been implemented. The recognition accuracy of the system using the proposed PPT achieved 77.23%. We also contrasted this method with previous conventional approaches to show its superior performance.

1. INTRODUCTION

In the past years, a number of researchers have proposed ways to use natural language modeling in automatic speech recognition, where it is usually implemented using the n-gram paradigm [1] and the grammatical constraints [2]. Nevertheless, previous workers had shown that conversational speech contains a significant number of disfluencies. For example, fillers such as *um* and *uh* are inserted frequently, as well as pauses, repetitions, and repairs. Therefore, the conversational speech is very different from read speech and these disfluencies dramatically degrade the performance of continuous speech recognition. Due to the effect of disfluent events, some phrases are undetected or misdetected and it will affect the estimation of the n-gram paradigm and the grammatical constraints. Recently, some conventional statistical language models are proposed to handle this problem [1][2]. However, the undetected/misdetected errors occur at not only the disfluent speech segment but also the fluent speech segment. Hence, their approaches are still unable to model the long distance relationships well.

In this paper, in order to deal with the problems of disfluencies, a PPT is proposed to recover undetected/ misdetected errors. Since the undetected/misdetected errors are generated from automatic speech recognizer, we assume that every functional phrase may be undetected/misdetected and can be skipped without affecting the main intention of the input speech. The generated phrase sequence is thus called a partial pattern of the original phrase sequence. According to these partial patterns, a statistical PPT is constructed to model the grammatical constraints. Thus, the long distance relationship can be modeled in the PPT. But in a training corpus, there are still many unobserved n-grams. In this paper a PPT merging algorithm, which is based on the word classes [3], is derived to deal with this problem.

2. SYSTEM OVERVIEW

The overall flow diagram of the proposed system is depicted in Fig. 1. In the training procedure, the functional phrase set is extracted automatically from the conversational speech corpus [4]. The key-phrase set can also be defined for a specific domain. Each internal node in the PPT represents a particular phrase and is characterized by two conditional distributions, which are modeled with and without undetected/misdetected errors respectively. Each external node in the PPT represents a partial pattern from its corresponding original patterns by skipping one phrase segment. In addition, the probability for the partial pattern generated from its corresponding original pattern is also included and used to estimate the score for error recovery. Finally, a PPT merging algorithm is proposed to reduce the number of partial patterns with similar syntactic structure based on minimizing an objective cost function.

In the recognition procedure, speech recognition is firstly performed to generate a phrase lattice. According to this phrase lattice, a Viterbi search algorithm is used to find the partial pattern candidates in the PPT. As an external node has been visited, a recovered sentence can be found by backtracking its corresponding original partial patterns. Therefore, the undetected/misdetected error from the speech recognizer can be recovered. Finally, the sentence verification is applied to re-rank the recovered sentences generated from the PPT.

3. PARTIAL PATTERN TREE

3.1 Partial Pattern

In general, each query sentence can be represented as a sequence of functional phrases and a key-phrase. Based on the observation, we assume that every functional phrase may be undetected/misdetected and therefore skipped. The sentence with one or more skipped functional phrases is then used to generate the partial patterns. Furthermore, in this paper each partial pattern is subject to contain a key-phrase in order to produce a semantically meaningful sentence. To apply our algorithm, it is sufficient to view a sentence, S_i, as a sequence of functional phrases and a key-phrase. Then it can be expressed as follows:

$$S_i = \left\{ FP_1^i, FP_2^i, \cdots, FP_{NB_i}^i, KP^i, FP_{NB_i+1}^i, \cdots, FP_{NB_i+NA_i}^i \right\} \quad (1)$$

where KP^i is the key-phrase and FP_i^j is the *j-th* functional phrase. NB_i and NA_i are the numbers of functional phrases before and after the key-phrase, respectively. A partial pattern is defined as a subsequence of phrases with the key-phrase. For instance,

"ABC" is a sentence where "A" and "C" are the functional phrases and "B" is the key-phrase. Then four partial patterns "ABC", "AB", "BC", and "B" can be obtained.

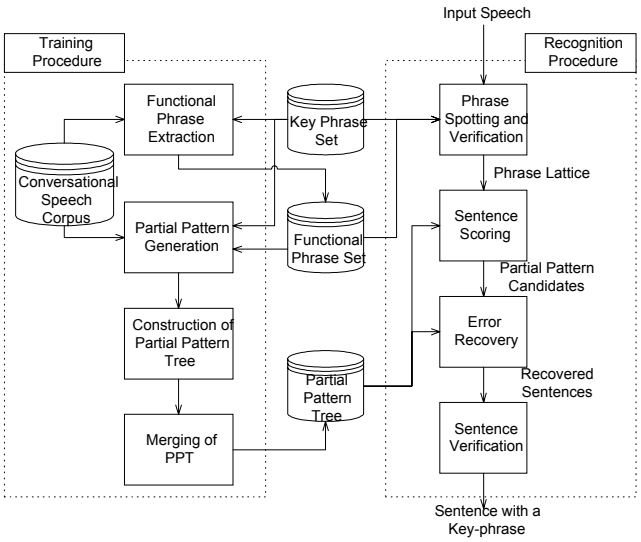

Fig. 1. Flow diagram of the proposed system

3.2 Construction of Partial Pattern Tree

Given a training corpus, all the query sentences can be decomposed into partial patterns, which contain the partial sentence grammar. Then, the partial patterns are used to construct the PPT. Finally, a maximum likelihood estimate is adopted to estimate the statistical relationship among phrases in the PPT and each external can be represented as follows:

$$EN_i = \{PP_i,\ Ptr_i,\ RPb_i\} \tag{2}$$

where PP_i is the corresponding partial pattern and the Ptr_i is the original pattern pointer set. RPb_i is the recovered probability of the original pattern pointer and used to find the possible error recovered sentences. Each internal node in the partial pattern represented as

$$IN_i = \left\{Ph_i, Fr_i, P\left(Ph_i \middle| Path_1^{L_i}\right), P\left(Ph_i \middle| FillerPath_1^{L_i}\right), Ns_i, Son_i\right\} \tag{3}$$

consists of six parameters such that

(a) Ph_i is the phrase in the functional phrase set or key-phrase set;
(b) FR_i is the phrase frequency of phrase Ph_i in internal node IN_i;
(c) $P\ (Ph_i|Path_1^{Li})$ is the conditional distribution of phrase Ph_i given the context history $Path_1^{Li}$ and L_i is the number of internal node before IN_i;
(d) $P\ (Ph_i|FillerPath_1^{Li})$ is the conditional distribution of phrase Ph_i given the context history $FillerPath_1^{Li}$;
(e) Ns_i is the number of sons of IN_i in the PPT;
(f) Son_i is the linked list of sons of IN_i in the PPT.

Each internal node in the PPT represents phrase Ph_j and is characterized by two conditional distributions $P(Ph_j|Path_1^{j-1})$ and $P(Ph_j|FillerPath_1^{j-1})$ given the context $Path_1^{j-1}$ and $FillerPath_1^{j-1}$. $Path_1^{j-1}$ and $FillerPath_1^{j-1}$ are the full length histories and shown as follows:

$$Path_1^{j-1} = Ph_1 Ph_2 \cdots Ph_{j-1} \tag{4}$$

$$FillerPath_1^{j-1} = Ph_1 Ph_2 \cdots Ph_{j-1} f \tag{5}$$

where f is the filler phrase used to represent the undetected/misdetected errors from speech recognition. Using the relative frequency estimator, the conditional distribution, $P(Ph_j|Path_1^{j-1})$ can be estimated as follows:

$$P\left(Ph_j \middle| Path_1^{j-1}\right) = P\left(Ph_j \middle| Ph_1 Ph_2 \cdots Ph_{j-1}\right) = \frac{C\left(Ph_1 Ph_2 \cdots Ph_{j-1} Ph_j\right)}{C\left(Ph_1 Ph_2 \cdots Ph_{j-1}\right)} \tag{6}$$

where $C(Ph_1,Ph_2...Ph_n)$ is the frequency of a certain n-gram, $Ph_1,Ph_2...Ph_n$, in the training corpus. In our approach, the functional phrase in each training sentence is presumed to be a filler phrase and used to train the condition distribution. Therefore, the conditional distribution with a filler phrase can be estimated as

$$P\left(Ph_j \middle| FillerPath_1^{j-1}\right) = P\left(Ph_j \middle| Ph_1 Ph_2 \cdots Ph_{j-1} f\right) = \frac{1}{\#FS_1^j(f)} \sum_{f^* \in FS_1^j(f)} P\left(Ph_j \middle| Ph_1 Ph_2 \cdots Ph_{j-1} f^*\right) \tag{7}$$

where $FS_1^j(f)$ is the set of functional phrases and f represents a functional phrase between Ph_{j-1} and Ph_j and can be treated as a filler phrase in the full length histories $Ph_1,Ph_2...Ph_{j-1}$ Ph_j .Therefore, the average probability with a skipped functional phrase between Ph_{j-1} and Ph_j is used to find the partial pattern candidates. There are three steps to construct the PPT. First, each sentence is converted into a phrase sequence. Second, the phrase sequence is decomposed into some partial patterns. Third, prefix search and new node creation are used to generate the new partial pattern path. Fig.2 shows an example of a PPT for the sentence "ABC" where "A" and "C" is the functional phrase and "B" is the key-phrase.

3.3 Phrase Merging

A PPT merging algorithm, which is based on the phrase equivalence class, is proposed to deal with the sparse data problem. The phrase equivalence class is the individual phrase group and can be automatically derived by minimizing a cost function. Consequently, the cost function, which is based on the distance between an infrequent phrase, Ph_i (the frequency of Ph_i is below a threshold) and a frequent phrase Ph_j (the frequency of Ph_j is above a threshold), should be described first and defined as follows:

$$CF\left(Ph_i, Ph_j\right) = \frac{1}{\#IN\left(Ph_i\right)} \sum_{IN(Ph_i) \in PPT} \delta\left(IN\left(Ph_i\right), IN_i\left(Ph_j\right)\right) \tag{8}$$

where $IN(PH_i)$ is the internal node with Ph_i in the PPT. Internal nodes with the same parent are called peers and $IN_i(PH_i)$ is the corresponding peer with respect to Ph_j in $IN(PH_i)$. $\delta(IN(PH_i), IN_i(PH_i))$ is the distance between internal nodes $IN(PH_i)$ and $IN_i(PH_i)$. The Kullback-Leibler distance measure [5] is adopted to measure the distance between two internal nodes and defined as follows:

$$\delta\left(IN(Ph_i), IN_i\left(Ph_j\right)\right)=\begin{cases}\sum_{x\in Vc} P_i(x)\log\left(\dfrac{P_i(x)}{P_j(x)}\right) & \text{if } IN_i\left(Ph_j\right) \text{ exists}\\ 1 & \text{otherwise}\end{cases} \tag{9}$$

where Vc is the union of key-phrase set and functional phrase set. $P_i(x)$ is the conditional distribution with the full history from root to node $IN(PH_i)$. Futhermore, the selected phrase Ph_j minimizing Eq. (8) can be defined as

$$Ph_j = \underset{Ph_k\in Vc, \#Ph_k>TH}{\arg\min}\ CF\left(Ph_i, Ph_k\right) \tag{10}$$

where $\#Ph_k$ is the frequency of Ph_k and TH is a predefined threshold. Therefore, the infrequent phrases Ph_i can be merged into phrase Ph_j. Using the phrase equivalence class, the PPT can be re-constructed.

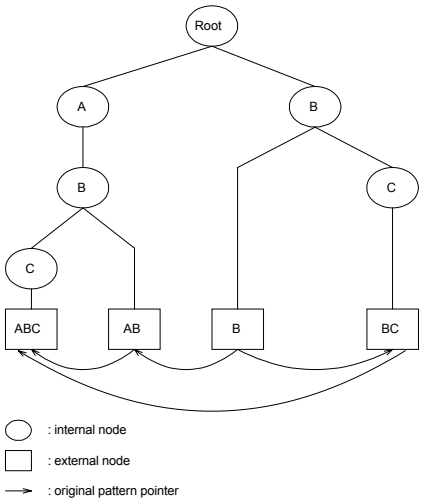

Fig. 2 An example of a PPT for the sentence "ABC" where "A" and "C" are the functional phrases and "B" is the key-phrase.

4. ERROR RECOVERY AND SENTENCE VERIFICATION

According to the phrase lattice and the PPT, the sentence-scoring algorithm based on the Viterbi search is applied to combine the verified key-phrases and functional phrases that the sentence hypotheses can be generated. The partial pattern candidates and the corresponding external nodes can be found and used to recover the undetected/misdetected errors resulted from speech recognition.

4.1 Error Recovery

Assuming that a partial pattern PP_1 has been determined, the skipped functional phrases can be sequentially found using a backtracking algorithm. Therefore, the basic idea of error recovery operation is to model the recovered process as a series of transformations between partial patterns and it can be defined as follows:

$$ERO\left(PP_1\right)=\left\{PP_1, ER_1, PP_2, ER_2, \cdots, PP_{N_{PP}-1}, ER_{N_{PP}-1}, PP_{N_{PP}}\right\} \tag{11}$$

where ER_k denotes the *k-th* error recovery operation which is applied to recover PP_k into PP_{k+1}. In order to determine the best operation sequence for error recovery, the likelihood value of a partial pattern with N_{ER} error recovery operations is computed as follows:

$$\begin{aligned}P\left(ERO\left(PP_1\right)\right) &= P\left(PP_1, ER_1, PP_2, ER_2, \cdots, PP_{N_{PP}-1}, ER_{N_{PP}-1}, PP_{N_{PP}}\right)\\ &= \prod_{i=2}^{N_{PP}} P\left(PP_i, ER_{i-1}\middle|PP_1, ER_1, PP_2, ER_2, \cdots, PP_{i-2}, ER_{i-2}, PP_{i-1}\right)\\ &\approx \prod_{i=2}^{N_{PP}} P\left(PP_i, ER_{i-1}\middle|PP_{i-1}\right)\\ &\approx \prod_{i=2}^{N_{PP}} P\left(ER_{i-1}\middle|PP_{i-1}\right)\end{aligned} \tag{12}$$

The approximation of Eq. (12) is based on the assumption that the segmentation and results only depend on the previous partial pattern, and PP_i is uniquely determined by ER_{i-1} and PP_{i-1}.

4.2 Sentence Verification

Suppose the error recovered sentence candidates, $ERO(PP_i)$, consists of N_{PPi} phrases, the confidence measure of $ERO(PP_i)$ is defined as a function of its phrase likelihood ratios. We have investigated two functional forms of the confidence measure. The first confidence measure, CM_1, is simply an average of log likelihood ratios of all the phrases and defined as

$$CM_1\left(ERO\left(PP_i\right)\right)=\frac{1}{N_{PP_i}}\sum_{j=1}^{N_{PP_i}} D\left(O_{i,j}; Ph_j^i\right) \tag{13}$$

where $O_{i,j}$ is the observation of the *j-th* phrase in partial pattern PP_i, $D(O_{i,j};Ph_j^i)$ is the log likelihood ratio of phrase Ph_j. The second one, CM_2, focuses on more confident phrases rather than averaging all the phrases. This is because some phrases in the error-recovered sentences may be misdetected. In order to reduce the effect of these errors, the confidence measure CM_2 can be defined as

$$CM_2\left(ERO\left(PP_i\right)\right)=\frac{1}{N_{PP_i}}\sum_{j=1}^{N_{PP_i}}\begin{cases}D\left(O_{i,j}; Ph_j^i\right) & \text{if } D\left(O_{i,j}; Ph_j^i\right)>0\\ 0 & \text{otherwise}\end{cases} \tag{14}$$

The final sentence verification uses the global acoustic and partial sentence grammar information on the entire input utterance. Therefore, the final sentence verification can be shown as

$$SV\left(PP_i\right)=\beta Score(PP_i, PL, PPT)+(1-\beta)CM\left(ERO\left(PP_i\right)\right) \tag{15}$$

where β is a weighting parameter, PL is the phrase lattice generated by speech recognizer, PP_i is the partial pattern estimated from a sentence scoring algorithm, and $ERO(PP_i)$ is the error recovered sentence candidate.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance, 2583 faculty names and 39 department names in National Cheng Kung University, Taiwan, were selected as the key-phrase set. We also recorded 3973 utterances spoken by a different group of 43 speakers (24 males, 19 females) for testing.

In the first experiment, the performance for phrase spotting was evaluated and the key-phrase error rate was 33.12%. The experimental results of two sentence verification strategies and weighting factor, β, are shown in Fig. 3. When β is 0.4, it is clear that the CM_2 outperformed other sentence verification

strategies and achieved a key-phrase error rate of 21.47%. This is because the misdetected errors have been considered in CM_2.

In the next experiment, the error recovery approach is adopted and the results are shown in Table I. We can find that the key-phrase error rate can be reduced slightly from 22.8% to 22.1%. In order to eliminate the problem of sparse data, the PPT merging algorithm is adopted and the results is shown in the Fig. 4. In Fig. 4, a specified phrase set, which is merged by human, is also used for comparison. Finally, Using different language models, the experimental results are shown in Table II.

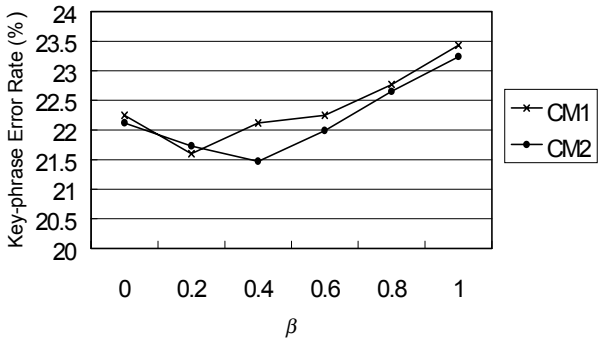

Fig. 3 The experimental results of two sentence verification strategies as a function of the weighting factor β

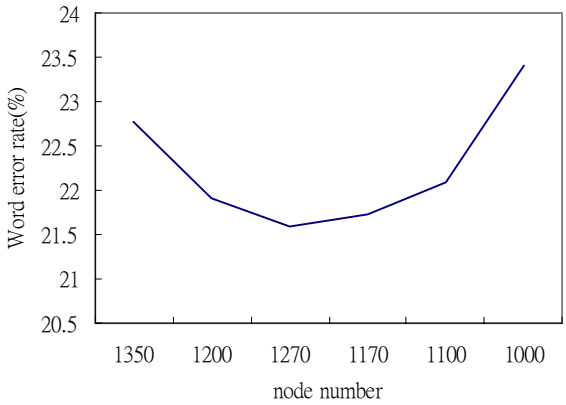

Fig. 4 The experimental results as a function of the number of nodes in the PPT

In addition to recognition experiments, we also conducted experiments where the main objective is to illustrate improvements in the testing sentence with out-of-grammar sentences. Another 742 testing sentence were recorded for testing and the experimental result is listed in Table III. We observe that the performance of the system using the PPT is better than that of other approaches. The main reason is that the partial sentence grammars in the out-of-grammar sentences can be modeled in the PPT.

6. CONCLUSION

In this paper, we have proposed a new method for conversational speech recognition. In order to deal with the undetected/misdetected errors, we assume that every functional phrase can be skipped without affecting the main intention of the input speech. According to this feature, a PPT is constructed and the statistical language modeling of the PPT is estimated. In the PPT, the long distance relationship between phrases can be also modeled. Finally, the sentence verification is also proposed to reduce the effect of undetected/misdetected errors and re-rank the recognition results. For the experiments, the results show that the proposed error recovery and sentence verification using PPT give an encouraging improvement.

7. ACKNOWLEDGEMENT

The authors would like to thank the National Science Council, Republic of China, for financial support of this work, under Contract No. NSC89-2614-H-006-003-F20.

Table I. The experimental results for adopting error recovery

	Key-phrase Error Rate (%)
Phrase Spotting	33.12
PPT without Error Recovery	22.8
PPT with Error Recovery	22.1

Table II. The experimental results for the system using different language models

	Key-phrase Error Rate (%)
Phrase Spotting	33.12
Phrase Spotting and Bigram	26.64
Phrase Spotting and Variable N-gram	24.48
Phrase Spotting and Proposed PPT	22.77

Table III. The experimental results for out-of-grammar sentences

	Key-phrase Error Rate (%)
Phrase Spotting	34.03
Phrase Spotting and Bigram	26.77
Phrase Spotting and Variable N-gram	24.6
Phrase Spotting and Proposed PPT	22.83

8. REFERENCES

1. Siu, M. and Ostendorf, M. (2000), "Variable N-grams and Extensions for Conversational Speech Language Modeling," IEEE Trans. on Speech and Audio Processing, 8, 63-75.
2. Guyon, I. and Pereira, F. (1995), "Design of a Linguistic Postprocessor Using Variable Memory Length Markov Models," in Proceedings of 3rd ICDAR, 454-497.
3. Niesler, T. and Woodland, P. (1999), "Variable-length Category N-gram Language Models," Computer Speech and Language, 21, 1-26.
4. Lai, Y. S. and Wu, C. H. (1999), "Unknown Word and Phrase Extraction Using a Phrase-like-unit-based Likelihood Ratio," in Proceedings of ICCPOL99, 1, 5-9.
5. Jelinek, F. (1998), "Statistical methods for speech recognition," Cambridge, MA: MIT Press.