A Comparison of Three Language Models for Speaker-Dependent Chinese Speech Recognition

Wing-Kwong Wong[1], Chien-Hsing Wu[2], Hsiang-Yen Chen[1], Chih-Tsun Chen[1], Sheng-Cheng Hsu[1]

[1]Institute of Electronic and Information Engineering

National Yunlin Universityof Science & Technology, Touliu, Taiwan 640

[2]Dept. of Electrical Engineering, National Chung Cheng University

[1]g9010802@yuntech.edu.tw

[1]Tel: 05-5342601 ext: 4391

Abstract

In this paper, we present a speech recognition system consisting of a signal processing component and a language model. The signal processing component uses traditional techniques such as linear precedence coefficient (LPC), Cepstrum, VQ and HMM. The three language models used in this study are those of context-free grammar, bigram and HMM. The target speeches to be recognized are Chinese Logo (CLogo) program codes. Since CLogo is a formal programming language, we can write down its context-free grammar rules by hand. For bigram model and HMM, their probabilistic linguistic knowledge are automatically learned from a corpus of CLogo programs. All three language models can output the best character combination for the spoken sentence. Empirical results show that CFG performs better than others, with a weakness is that it must be constructed by hand. On the other hand, HMM and bigram can be trained automatically with a corpus. The system is also tested with a corpus of natural language texts of elementary geometry problems.

Keywords

Chinese speech recognition, language models, context-free grammar, Hidden Markov model, bigram

Introduction

With the advancement of computer technologies, many tasks can be assisted by computer, e.g. computer-assisted learning. For example, kids can write CLogo [1] programs to draw geometric figures on a computer. However, when communicating with a computer, they still have to depend on a keyboard, or a mouse to input commands. Especially for children at grade one and two, they might find it difficult to enter commands when using a computer. If the computers can recognize their speech, they can give orders to a computer verbally.

In this article, we describe a speech recognition system to help children to write CLogo programs. We use some basic digital signal processing techniques and several language models to increase the recognition rate.

With regard to speech signal processing, we first divide an input speech into signals of single character by using energy measure and zero-crossing rate. Then the speech signal of every character sound is divided into a number of frames to extract the features of the speech signals and a set of feature values are computed for each frame. Finally, with the technique of vector quantization, the feature of every frame is compressed into a codeword. A series of codewords make up a codebook. During the training phase, the codebook is used to build Hidden Markov Models (HMM). For testing, the trained HMMs are used to determine the "nearest" model as the recognized character.

After the basic signal processing of every character of the input sentence, a language model is used to select the best combination of candidate characters for the sentence. The three language models we use are those of context-free grammar, bigram and HMM. Since CLogo is a formal programming language, we can write down its context-free grammar rules by hand. For bigram model and HMM, their probabilistic linguistic knowledge is learned automatically from a corpus of CLogo programs. All three language models output the best word combination for the input sentence. In discussion, We use MSA method for increasing recognition rate and now is in proceeding.

1. Literature Review

Chinese speech recognition have been studied extensively. Wang [2] established a model of speaker-independent Mandarin tone recognition using VQ and HMM techniques. Li [3] introduced a Mandarin speech model (Voice Dictation of Mandarin Chinese) using Isolated-Syllables to construct a HMM. Chien [4] built a novel framework of an online unsupervised learning algorithm to flexibly adapt the existing speaker-independent hidden Markov models (HMMs) to nonstationary environments induced by varying speakers, transmission channels, ambient noises. Hon [5] developed SPHINX, which is an HMM-based recognizer using multiple codebooks of various LPC derived features. In these and other studies, VQ-HMM has been applied broadly and successfully for Chinese speech recognition. Statistical language models can play an important role in continuous speech recognition, but their performance is often unstable due to the sparsity problem of the training data. Lee [6] reported that SPHINX achieved a word accuracy of 53.4%, using bigram grammar or Word-Pair, and bigram had better performance than Word-Pair grammar. Ma [7] proposed the approximation that the probability of word depends on only the immediately proceeding word in bigram models. Bin [8] reported similar results. For Chinese language models, HMM and

context-free grammar are seldom used. In this paper, we compare these two models and the bigram model using results from speech recognition experiments.

2. System architecture

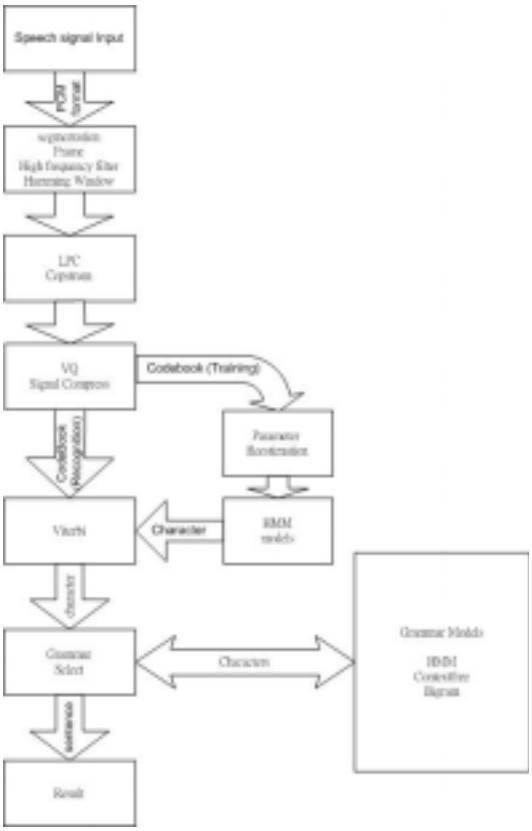

Fig. 1 System architecture

2.1 Speech Signal Input

Speech data were recorded using 8 bit unsigned and sampling rating at 8000Hz and stored in PCM format. We use PCM because its ASCII content can be viewed by regular text editor, and it is simple to combine PCM files, do scaling, and word signal analysis

Since the original speech signal data take up a lot of disk space and are difficult to be compared directly for recognition, we must extract characteristic features of the speech signals that can be used for recognizing new input speeches.

2.2 Phonetic Segmentation Process

First of all, we divide the input speech signals into individual character signals. For each character signal, the feature parameters are extracted. For Chinese speech, one sound is generally one Chinese character. Thus a series of speech signals are divided into each character's phonetic signals. The parameters of short-time energy and zero-crossing rate are used for separating speech signals.

2.3 Feature Extraction

If speech signals are compared to the learned signals directly, there are large amounts of data and the processing time will be very slow. Therefore, we must extract the characteristic features, which can be used to compare to the features of the characters that have already been learned. Though it is not likely that signals of the same speech are identical, but the characteristics of similar speeches would be similar. A feature extraction method is to divide the phonetic signals into frames, each of which would produce a characteristic feature. Some loss of high frequency is compensated and Hamming Window is also used. Finally, a set of Linear Predictive Coefficient (LPC) [9] is computed and transformed to Cepstrum coefficients as the features of the input speech.

2.4 Signal Vector Quantization

After the speech signals are trained into Cepstrum coefficients, these features are compressed further with the technique of vector quantization [10][11]. This will increase the recognition rate. The codebooks are trained with the LBG algorithm [10]. We have limited the size of codebooks to 64. For each frame, one codeword is output from each vector quantizer and is made available for the HMM stage.

2.5 Hidden Markov Model

Once the input features have been quantized, a HMM [12] is trained for each character. Figure 2 shows the type of HMM we are considering here. The model is based upon a left-to-right Markov chain, which starts at state 1 and ends at state 4. This model has already demonstrated success for recognition of isolated digits.

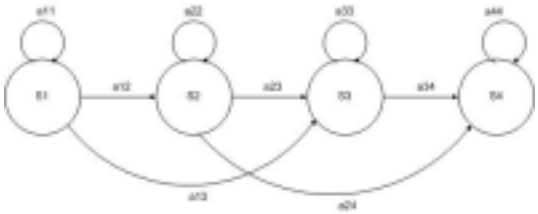

Fig. 2 Hidden Markov Model

3. Language Models

For the speech recognition method at the signal processing level as described above, the recognition rate is not very good. One major drawback is that each character of the input sentence is recognized individually. That is, the choice of one character does not depend on its surrounding characters. This lack of linguistic knowledge can be compensated with a language model. After a speech is entered as input, the signal recognition component will choose the top seven candidates for the speech signal of each character. Then we use a language model to pick out an optimal combination of the candidates of each character so that the resulting sentence is a most likely sentence for the input speech signals. For our application, the target speech is a CLogo program. We try to compare the pros and cons of three language models for recognize speeches on CLogo programs.

3.1 Context-free Grammar

A context-free grammar defines the possible syntactic structures of a language. For our application. We use a chart parser to determine whether a sentence is legal or not [13].

A partial grammar for the CLogo language is given in Figure 3.

```
S->EXP  S
S->EXP
EXP->I
D-> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
N->D
N->D  N
F->D
F->D  F2
F2->N2
N2->D
N2->D  N2
I->前  進
I->前  進  N
  …
I->加  F  F
```

Fig. 3 Grammar of the CLogo language

For the grammar, “ I ” stands for an instruction. For example, the rule “ I -> 前

進 N ” means “ 前 進 ” can take a numeric argument ; “ I -> 加 F F ”, and the operation of “ 加 ” needs two floating numbers as its arguments.

We work out one example to explain how to use the parser to improve the result of speech recognition. As we input the speech “前進 149”, the signal processing component will determine the top three candidates for each character[1].

Table 1. Top three candidates of five characters

score	3	2	1
W1	弦	前	顯
W2	進	進	印
W3	低	低	1
W4	4	置	平
W5	9	首	否

Without any grammatical restriction, the best recognition result will “弦進低 49”, which is not a legal sentence in CLogo. In a regular syntactic parser, a character of an input sentence might have different syntactic categories, e.g. “fly” can be a noun or a verb. The parser will pick a category for “fly” automatically during parsing. Since our parser output all possible parse trees, the “fly” might have different categories in different parse trees. For the speech recognition purpose, a character’s candidates (determined by the signal processing component) are processed as the possible “syntactic” categories of the word. The parser will then take care of the rest. For the above example, two parse trees result, improving the recognition speech as “前進 1 置首” or “前進 149”

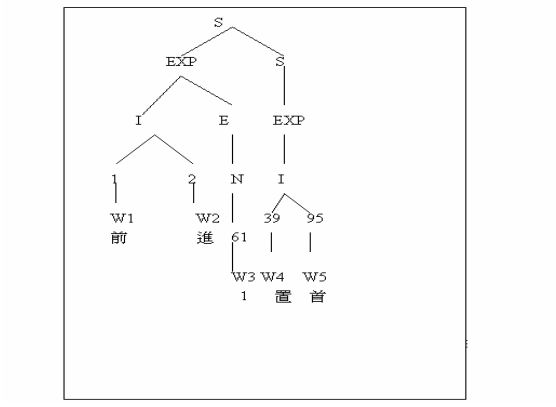

Fig.4 Parsing Tree 1 as a result of CFG model

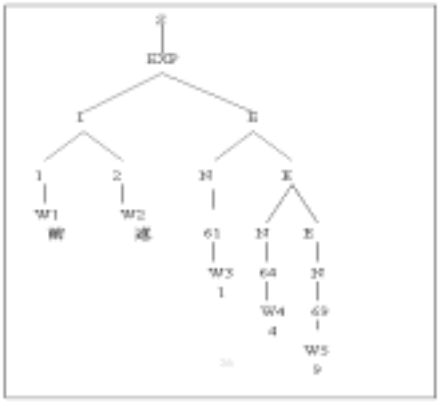

Fig. 5 Parsing Tree 2 as a result of CFG model

3.2 Bigram

The second language model is the bigram model. Instead of using the predefined grammar rules as in the CGF model, we can determine the relationship between characters by computing the probability of composition of characters. The probability of N-characters string is estimated as bigram probabilities [13].

[1] The system actually uses the top seven candidates instead of three. Three candidates are used here for illustration.

$$P(C_1, C_2, ..., C_N) \approx \prod_{i=1}^{N} P(C_i \mid C_{i-1})$$

For the two characters, C_i and C_{i-1}, we compute the bigram probability $P(C_i \mid C_{i-1})$ from characters and the given corpus. The higher the frequency that C_{i-1} and C_i occur consecutively, the greater $P(C_i \mid C_{i-1})$ is. $P(C_i \mid C_{i-1}) = 0$ means that there is no consecutive occurrence of C_{i-1} and C_i in the corpus. But this does not mean that the composition of C_i and C_{i-1} will never occur in future tests. So there is a need to correct the probability $P(C_i \mid C_{i-1}) = 0$ by replacing the value with a tiny number. This is called smoothing [14]. The probability of $P(C_i \mid C_{i-1})$ is:

$$P(C_i \mid C_{i-1}) = \frac{count(C_{i-1}C_i)}{count(C_{i-1})}$$

In the bigram probability table, each voice has seven alternative characters. We can find the best composition of phase with Viterbi algorithm and bigram. An example of an input speech "方向" is shown in figure 6. Without using the bigram, the speech is recognized as "方消". The bigram model produces the correct recognition.

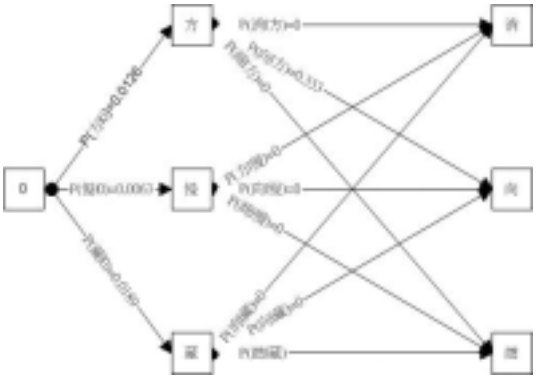

Fig.6 Bigram and Viterbi algorithm

3.3 Hidden Markov Model

With the bigram model, we estimate the probabilities of the two consecutive characters, but the amount of information for all possible bigrams is huge if there us a large vocabulary. So less expressive model is to use a language model. The third language model, a less expensive approach, is to use HMM to model the association of adjacent characters. After the signal word, processing model produces the top seven the best composition of characters is chosen with Viterbi algorithm and the learned HMM. The HMM uses four states, and each state can output a symbol from 160 character candidates 160 symbols.

4. Experiment Results

For single character recognition at the signal processing model, we compare the results of using VQ alone and using VQ + HMM. Since there are 160 characters and a HMM model is trained for each character. The result is given in Table 2, showing that HMM can improve the recognition result when used with VQ.

Table 2 :

Method / Samples	VQ	VQ-HMM
160	83.5%	85%

For continuous speeches, the sound of each character will depend a lot on the meanings of the words in the speech, the rate of recognition for continuous speeches will be much lower than for individual characters, so this research use different language models to improve the rate of

recognition.

For our experiments, 101 training patterns and 202 testing patterns are used. A pattern is a continuous speech of a short phrase. The top seven candidates of each character is determined by the signal processing component. Then three different language models are used to improve the recognition result:

Experiment one: No language model.
Experiment two: Context-free grammar.
Experiment three: Bigram model.
Experiment four: HMM

Table 3. Experiment Results

Training pattern / Testing pattern	Exp 1 No Language Model	Exp 2 Context-free Grammar	Exp 3 Bigram	Exp 4 HMM
101/202	68%	98.08%	96%	90%

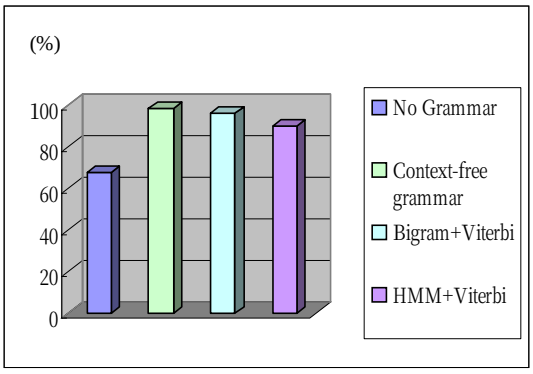

Fig 7. Bar chart of experiment results.

4.1 Comparing Bigram and HMM recognition error.

For HMM, the probabilities of "移", "至" appearing as the first or second character of a phrase are greater those of "停" and "止", so HMM prefers "移至". For the bigram model given P(停|0) = 0.006329, P(止|0) = 1. Viterbi prefers the two words "停止", since its path is optimal.

Table 4 Comparing Bigram and HMM (1)

Target	Candidate words	HMM	bigram
停	停０移示列低弦	移	停
止	指止至置次字整	至	止

For the target "移至座標", the bigram model succeeds while HMM fails to recognize it. HMM produces the result of "顯示軸標" even though it is not a legal CLogo instruction.

Table5 Comparing Bigram and HMM (2)

Target	Candidate words	HMM	bigram
移	１移記音顯 b 示	顯	移
至	置至止字４指示	示	至
座	縮否左縱座軸示	軸	座
標	消標右角示框空	標	標

From the corpus the following bigram probabilities are obtained: P(整 |0)=0.02531, P(數 | 整)=1, P(亂 |0)=0.00632, P(數|亂)=1. The probability of the character "整" at the beginning of a sentence is greater than that of the character "亂", so the bigram model prefers "整數". In the corpus, "整數" occurs four times, but "亂數" occurs only once. HMM produces the correct recognition.

Table 6 Comparing Bigram and HMM (3)

Target	Candidate words	HMM	Bigram
亂	反亂然示轉整商	亂	整
數	一組數鼠除讀出	數	數
2	２格高個　示藏	2	2

.	. 點顯列變邊示	.	.
3	3 函慢錄藏示橫	3	3

The corpus has the following bigram probabilities: P(整 | 空)=0.025316, and P(則|否)=0. There is no training pattern "否則" in the corpus. So the bigram prefers the other choice of "空整", even though neither "假如空整" nor "空整" are legal CLogo instructions.

Table 7 Comparing Bigram and HMM (4)

Target	Candidate words	HMM	bigram
假	假角高長消樣示	假	假
如	錄如讀果出左除	如	如
否	否示－落或尾空	否	空
則	則整格程乘商示	則	整

In general, show experiment results, HMM does not learn the association between adjacent characters very well. For example, consider the target speeches of "前進" and "顯示". P(前進) = P(前) * $P(S_1->S_2)$ * P(進) = 0.404 * 0.42 * 0.374 = 0.0062, and P(顯進) = P(顯) * $P(S_1-S_2)$ * P(進) = 0.437 * 0.42 * 0.374 = 0.0067. The Viterbi algorithm would pick "顯進" as the optimal path even though "顯進" is not a legal CLogo instruction.

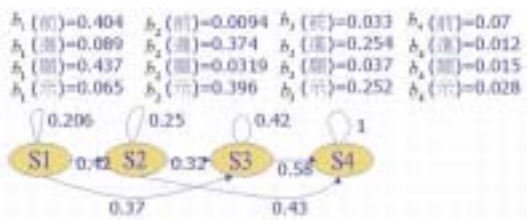

Fig. 8 HMM

We conclude that for the HMM, the probability of a character's occurrence at the Nth state indicates strongly probability that the character occurs in the Nth position of a phrase. On the other hand, the HMM model does not indicate the association strengths of adjacent characters, which are best represented by the bigram probabilities. This bias of the HMM model is corrected by the bigram model. Since P(前進)=1 >> P(顯進) = 0, due to the fact that "前進" is a CLogo instruction while "顯進" is not.

A problem with the CFG model is that of multiple parses. For the target speech "停止", the "停 0 移示列低弦" are the top seven candidates of the first character and "指止至置次字整" are these of the second character. There are two possible character combinations: "移至" and "停止". If in the training corpus, the probability of "移至" is bigger than that of "停止", an incorrect answer will be obtained.

4.2 Geometry Corpus with a Bigram Model

We not only use a bigram model on the CLogo corpus but also on natural language problem sets about geometry at elementary and junior high school level. There are 390 different characters and totally 5738 characters in the training geometry corpus. We build a bigram model and test the model with the texts of three geometry questions. In Question 1, which is 44 characters long, is "已知冒號 P 為線段 AB 垂直平分線上的任一點逗號 O 垂直平分線與線段 AB 的焦點求証 PA 等於 PB". Question 2 has 85 characters and Question 3 has 53 characters.

For both the CLogo corpus and the geometry corpus, the bigram model achieves recognition rate above 90% above. Note that the training corpus of geometry texts is large than that of CLogo.

Table 8. Recognition rate of geometric questions.

	Question 1	Question 2	Question 3
Length of the question	44	85	53
Recognition rate	88.89%	97.65%	100%

5. Discussion

We propose to induce rule templates automatically from corpus, and then use the rule templates as a language model to increase the speech recognition rate.

First, we build a table that contains the classified results based on the semantic/syntactic categories of words. For example, we classify “三角形”, “四邊形” as a geometric object (go), and “底邊”, “邊長” as a geometric attribute (ga), etc. We tag the words in the corpus, and then induce rule templates from the corpus.

Table 9. Classification of words.

go	Ga	num	…	conj
三角形	同位角	1	…	也
四邊形	半徑	2	…	及
…	…	…	…	…
直線	重心	九	…	因為

Table 10. Rule templates.

No.	Rule template
1	pun num pun
2	num ga adj poss ga
…	……
n	go num ga gr ga gr num quan

Since the signal recognition component will choose the top seven candidates for each character, we find every word/phrase from each four word candidates and use the words/phrases found for parsing.

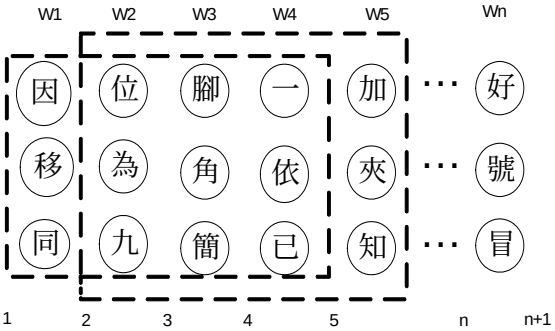

Fig. 9 Consider word/phrase of maximun length four.

Table 11. Parsing Chart.

Start position	Word/Phrase	End position	Category
1	因為	3	*conj*
1	同位角	4	*ga*
…	…	…	…
4	已知	6	*given*

Now we can use the words/phrases for parsing. For example, in Table 11, we found “同位角” is a word and its end position is 4, we search for next word/phrase whose start position is 4, in Table 11 is “已知”, and so on. If [***conj given***] is a legal rule template, then we add the new phrase, [***conj given***] is

added to the chart and used for further parsing. Using the words/phrases in the chart table, we can obtain a good parse with a maximum score among all parses. Exactly how the scores of each word/phrase is computed and how to induce the rule templates automatically from a corpus are being investigated. This method is similar to parsing with CFG yet we do not need to write the grammar rules by hand.

Conclusion

In this paper, we present a speech recognition system consisting of a signal processing component and a language model. The signal processing component uses traditional techniques such as linear precedence coefficient, Cepstrum, VQ and HMM. Speech recognition based on the signal processing component alone suffers from its ignorance of lexical association and syntactic structure of the target speeches. This problem is addressed by the addition of a language model. In particular, this paper considers three models and compares their strengths and weaknesses. These models are context-free grammar, bigram, and HMM. The CFG model is the theoretically and practically the best since our target speeches are CLogo instructions, which can be formally defined with grammar rules. Its major problem is that of multiple parses, and we propose a scoring scheme to select the best parse. Another problem is the cost of constructing grammar rules by hand, especially when there are many grammar rules with a large vocabulary. Most importantly, this technique cannot be directly applied to natural language speeches, since it is very difficult to construct a CFG, especially for the Chinese language.

The bigram model seems to be the next best bet after CFG. This model can be used for any kind of speech, including formal or natural languages. As long as we can collect a reasonably large and representative corpus, the bigram model can be constructed automatically, at least in theory. In practice, however, for a language with a large set (size N) of vocabularies, the space requirement of a full bigram model is N^2, which poses a practical problem. If N is large, say 10^3, then N^2 is 10^6. The worst model among the three is HMM. This model's weakness is its relative ignorance of the association strengths of adjacent words, which can be captured well by bigram model. But its weakness might be due to the particular configuration of our HMM model. Other configurations might perform better. Moreover, HMM does have some practical advantages. First, it can be trained automatically with a corpus, where a CFG must be constructed by hand. Second, its memory requirement, in our configuration, is much cheaper than that of the bigram model.

Reference:

[1] Wong, W. K., Chan, T. W., Pai, S. T., Wang, Y. K., Chen, Y. S., and Hsu, W. L., "Natural Language Educational Agents in a Networked Chinese Logo Learning Environment", *Proceedings of ICCE*, Vol. 1, AACE, Beijing,

pp. 220-227. , 1998

[2]Yang, W.-J., Lee, J.-C., Chang, Y.-C.and Wang, H.-C. “Hidden Markov model for Mandarin lexical tone recognition”, *Proceedings of IEEE Transactions on Acoustics, Speech and Signal Processing*, Volume 36 , Issue 7 ,July 1988, Page(s): 988 –992.

[3] Lee Lin-Shan,“Voice Dictation of Mandarin Chinese” *IEEE Signal Processing Magazine,*Volume: 14 ,Issue:4, July 1997, Page(s): 63 –101

[4] Chien, J.-T. “Online unsupervised learning of hidden Markov models for adaptive speech recognition”, *IEE Proceedings-Vision, Image and Signal Processing* ,Volume: 148 ,Issue: 5 ,Oct. 2001, Page(s): 315 –324

[5] Lee, K.-F., Hon, H.-W., Hwang, M.-Y., Mahajan, S.and Reddy, R. “The SPHINX speech recognition system” , *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-89)* ,1989 , Page(s): 445 -448 vol.1

[6] Lee, K.-F and Hon, H.-W. “Large-Vocabulary Speaker-Independent Continuous Speech Recognition Using”, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP-88),*1988, Page(s): 123 -126 vol.1

[7] Ma Jiyong , Gao Wen, Wu; Jiangqin and Wang Chunli “A continuous Chinese sign language recognition system”, *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000 ,Page(s): 428 –433

[8] Tian Bin, Tian Hongxin, Fu Qiang and Yi Kechu “A novel statistical language modelling method for continuous Chinese speech recognition”, *Proceedings of Fourth International Conference on Signal Processing (ICSP '98)*, 1998, Page(s): 734 -737 vol.1

[9] Hayes Monson H, *Statical Digital Signal Processing And Modeling*,

[10] Huo Qiang and Chan Chorkin, “Contextual Vector Quantization for Speech Recognition with Discrete Hidden Markov Model”, 1994

[11] Lai,Jim Z.C. and Lve,C.C., *“Fast Search Algorithms for VQ Codebook Generation”*, PP163-168, 1996

[12] Rabiner Lawrence R , *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, 1989

[13]Allen James, *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc. 1995,

[14] Manning C.D. and Schutze H., *Foundations of Statistical Natural Language Processing*, The MIT Press, Massachusetts, USA, 2001