Linguistic Constraints

For

Large Vocabulary Speech Recognition

A thesis presented

by

Roger H.Y. Leung

to

The Electronic Engineering Department

in partial fulfillment of the requirements

for the degree of Master of Philosophy

The Chinese University of Hong Kong

Hong Kong

June 1999

Abstract

Over the past decade, significant progress has been made in speech recognition. Current state-of-the-art Chinese speech recognition systems are capable of achieving character accuracy of 80% on continuous speech recognition tasks using 29,000 words (from IBM, 1996). While the size and performance of modern speech recognition and understanding systems are impressive, current approaches to continuous word recognition utilize little linguistic knowledge in phonological, lexical and syntactic level. We believe the use of phonological and lexical knowledge through lexical access, as well as syntax knowledge through language model would be beneficial to speech recognition.

This thesis presents an algorithm for the construction of lexical access model that attempts to speed up a large vocabulary isolated word recognizer. Additionally, we describe different Chinese language modeling technique of a large vocabulary system at the character level and word level. These language models provide a powerful constraint to the recognizer. Finally, different kinds of n-gram smoothing methods are studied, with the aim of solving the problem of uneven distribution.

The results of this thesis support the argument that linguistic knowledge is beneficial to speech recognition.

Keywords:

Speech Recognition, Chinese, Lexical Access, Language Model, Smoothing

摘要

過去十年，語音科技的發展迅速，現時最先進的中文語音識別系統應用於29000字的連續性語音識別任務中可以達到80%的準確性(來源：萬國商業機器，1996)。雖然現代語音識別系統的規模和表現已經令人嘆爲觀止，但是現時的語音識別處理方法較少利用到語言學裏的音韻、詞滙和語句結構的知識。我們相信一方面利用音韻和詞滙的知識來選取詞滙，另一方面使用語句結構的知識來製作語言模型會對語音識別有幫助。

本論文介紹了詞滙選取的算法。這方法嘗試去加快大詞彙、單詞識器的運作。另外，我們描述了語言模型用於大詞彙的識別器。這模型對識別器提供了一個有效的語言約束。最後，我們研究了不同的平滑化語言模型，它們用於解決多聯模型(n-gram)不平均分佈的問題。

本論文的結果支持了語言學能夠幫助語音識別的論點。

關鍵詞：

語音識別、中文、詞滙選取、語言模型，平滑化

Acknowledgements

The completion of this thesis would not have been possible without the help of many people to whom I would like to express my heartfelt appreciation.

Firstly, I would like to express my deepest gratitude to my supervisor Prof. Hong C. Leung for being everything a supervisor should be. He taught me not only speech recognition, but also about thinking and behaving like a scientist. Without his great enthusiasm and continuous encouragement, the work would not have been completed.

Another person whom I would like to thank is my friends, Finoa Tse and Karen Ip, who shared with me their expertise in computational linguistics and proof reading.

Thanks are also given to the department for providing me with excellent facilities and environment for my research. I received much support from my colleagues, such as K.F. Chow, Y. Chen, C.Y. Choy, C.P. Chan, W. Lau, Y.W. Wong, etc… at the DSP laboratory. I am also grateful for the financial support from Sir Edward Youde Memorial Fund.

Another excellent person whom I would like to thank is my friend, Desiree Lam, for supporting me, and for encouraging me to do my best at what I choose to pursue.

Finally, my biggest gratitude is undoubtedly to my fiancée, Yvonne Lee. Her love and unwavering support were my continuing sustenance.

Table of Contents:

Table of Figures:

Table of Tables:

Chapter 1 Introduction

In this thesis, we describe novel techniques to solve the large vocabulary problem of speech recognition. We investigate the usefulness of lexical access for large speech recognition. Moreover, we investigate problems of building probabilistic language models for Chinese.

In this chapter, we describe the popularity of Chinese in the world's languages. First, we discuss specific problems for Chinese speech recognition. Secondly, we explore different kinds of knowledge that can be used. Finally, we identify our target research area of Chinese speech recognition.

Chapter 2 introduces fundamental linguistic knowledge, which would be useful throughout this thesis. Chapter 3 describes a method of lexical access by broad classes features. Our analysis of Mandarin broad class is compared with English and Cantonese. In chapter 4, two different kinds of language modeling approaches are studied. They are character level language model and word level language model. Smoothing methods for improving the language models are introduced in chapter 5. Four different kinds of smoothing techniques are compared. They are Witten-Bell, Good-Turing, absolute and linear smoothing. Detailed algorithm and its underlying inspiration are also presented. An experimental continuous Mandarin speech recognizer is also developed. Chapter 6 presents the conclusion of this thesis.

1.1 Languages in the World

Mandarin is the most popular language in the world. There are 864 million people speaking Mandarin. Cantonese is also a popular Chinese dialect. It is spoken in the southern provinces of Guangdong and Guangxi, Hong Kong and Macau, as well as throughout Southeast Asia in such places as Singapore, Malaysia, Thailand and Vietnam. There are 63 million Cantonese speakers. Table 1 shows the most popular languages in the world.

Family	Language	Geographic area	Rank	Number of speaker
Sino-Tibetan	Mandarin	North China	(1)	864,000,000
Indo-European	English	North America, Great Britain, Australia	(2)	443,000,000
Indo-European	Hindi	Northern India	(3)	352,000,000
Indo-European	Spanish	Spain, Latin America	(4)	341,000,000
Indo-European	Russian	Russia	(5)	293,000,000
Afro-Asiatic	Arabic	North Africa, Middle East	(6)	197,000,000
Altaic	Japanese	Japan	(9)	125,000,000
Sino-Tibetan	Cantonese	South China	(20)	63,000,000

Table 1: The data re-compiled from the 1992 World Almanac and Katzner. [1]

Although there are twice as many Mandarin speakers as English speakers, the development of Mandarin Chinese speech recognition is still lagging behind than that of English. In addition, the development of Cantonese speech recognition was not started until recent years. The motivation of this thesis comes from the huge potential need for Chinese speech recognition technology, and the relative lately development of the technology [3].

1.2 Problems of Chinese Speech Recognition

There are some technical reasons for the late development for Chinese speech technology. Chinese has her unique features, which are very different from western languages. The major obstacles for large-vocabulary Chinese speech recognition are listed below [4].

1.2.1 Unlimited word size:

There are about 10,000 commonly used Chinese characters. One to several numbers of characters can be combined to form a Chinese word. The combination of such characters gives an almost unlimited number of words, in which at least some 100,000 are commonly used and can be found in different version of dictionaries and texts on different subjects. Hence, it is extremely difficult to include all Chinese Words in a speech recognizer.

1.2.2 Too many Homophones:

Chinese words are formed by a combination of characters. Each character in turn maps to a syllable. The total number of phonologically allowed Mandarin tonal syllable is about 1,300. In other words, a limited number of syllables maps to a much larger number of monosyllabic characters. Hence, the problem of homonym is very severe. On the average, each Mandarin syllable is shared by about 7.7 (10,000/1,300) Chinese characters. This one-to-many mapping introduces many ambiguities in speech recognition.

The Chinese speech recognition algorithms must then be able to distinguish between Chinese homophones. In English, it is unusual to find three words which are homophones e.g. two, too and to. Homophones are much more common in Chinese. An analysis of Callhome Lexicon shows that only 85% of the Chinese words can be uniquely specific with Mandarin tonal syllables.

1.2.3 Difference between spoken and written Chinese:

There are many differences between spoken Chinese and written Chinese. It is rather surprising to notice that nearly 40% of the words used in a single case of court proceedings are not found in the

overall list of 43000 words used in Hong Kong newspapers for an entire year [6]. It reflects a vast gap between the language used by the Cantonese speakers in Hong Kong and the language they are expected to use in the context of written language, as found in newspapers. Mandarin Chinese has fewer discrepancies between its spoken and written forms. However, the problems still affect the performance of a speech recognizer, when its language model is trained on written text or when its acoustic model is trained on read speech.

1.2.4 Word Segmentation Problem:

While words in western languages are separated by white spaces, there are no delimiters between Chinese words. A language model is typically trained by segmented text. The segmentation ambiguity of training text in Chinese may hurt the frequency counts of the language model, and hence adversely affects the recognition results.

1.3 Different types of knowledge

In order to solve the unique problem of Chinese speech recognition, we propose to use additional linguistics information in the recognition process. Linguistics knowledge can be divided into phonetics, phonology, prosody, morphology, syntax, semantics and pragmatics. The acoustic model of a speech recognizer captures some of the phonological effects. Morphology, syntax, semantics and pragmatics could all be incorporated in the language model. On the other hand, speech knowledge can be classified into two dimensions [7]: the linguistic level of knowledge and its validity across different type of situations, such as prior knowledge, conversation-dependent knowledge and speaker-dependent knowledge. This classification is shown in Table 2. Most of the knowledge in the two lowest rows (parametric and phonemic) can be captured by the acoustic model. However, all the other types of knowledge could potentially be handled by lexical access and language model.

Type of Knowledge	Prior Knowledge	Conversation-dependent Knowledge	Speaker-dependent Knowledge
Pragmatic and Semantic	Prior semantic knowledge about the task domain	Concept sub-selection based on conversation	Psychological model of the user
Syntactic	Grammar for the language	Grammar sub-selection based on topic	Grammar sub-selection of the speaker
Lexical	Size and Confusability of the vocabulary	Vocabulary sub-selection based on topic	Vocabulary sub-selection and ordering based on speaker preference
Phonemic and Phonetic	Characteristics of phones and phonemes of the language	Contextual variability in phonemic characteristics	Dialectal variations of the speaker
Parametric and Acoustic	Prior knowledge about the transducer characteristics	Adaptive noise normalization	Variations resulting from the size and shape of vocal tract

Table 2: Different kinds of speech knowledge defined by Reddy D.R. et al. [7]

1.4 Chapter Conclusion

In the previous sections, the popularity of Chinese language is mentioned. In addition, the issues in Chinese speech recognition are highlighted. Furthermore, various types of speech knowledge are introduced, with the aim of solving the problems of Chinese speech recognition. In this thesis, our main emphasis is on the use of lexical and syntactic knowledge. Lexical access and language model will be employed to handle the problems.

Chapter 2 Foundations

2.1 Chinese Phonology and Language Properties

2.1.1 Basic Syllable Structure

Chinese is different from many western languages in that it is monosyllabic and tonal. While there are more than 10,000 monosyllabic Chinese characters, there are typically only about 1,300 tonal syllables in each of the Chinese dialects. Thus, many Chinese characters share the same pronunciation. However, often depending on the context, each Chinese character may have multiple pronunciations. Consequently, Chinese is a complex language with many-to-one and one-to-many mappings between the characters and the syllabic pronunciations. The notion of a Chinese word is also very different from many western languages. While the syllables and characters are relatively well defined, the Chinese words are composed of a variable number of characters. Since a Chinese word can be formed, in principle, by any combination of ~10,000 Chinese characters, the vocabulary of a speech recognition system can be huge.

Each tonal syllable can be considered as two independent parts, tone and base syllable. There are five lexical tones: 1) high-level tone, 2) mid-rising tone, 3) falling-rising tone, 4) high-falling tone, 5) neutral tone. Moreover each base syllable can be divided into Initial and Final parts. Table 3 and Table 4 list all Initials and Finals in Mandarin.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
/j/	/ch/	/sh/	/r/	/tz/	/ts/	/s/	/g/
8	**9**	**10**	**11**	**12**	**13**	**14**	**15**
/g/	/k/	/h/	/ji/	/chi/	/shi/	/d/	/t/
16	**17**	**18**	**19**	**20**	**21**	**22**	
/n/	/l/	/b/	/p/	/m/	/f/	null	

Table 3: The 22 Mandarin initials including null initial

Category	Member
1	Null
2	/a/, /ai/, /au/, /an/, /aŋ/
3	/o/, /ou/
4	/e/, /eh/, /ei/, /en/, /eŋ/, /er/
5	/u/, /ua/, /uo/, /uai/, /uei/, /uan/, /uen/, /uaŋ/, /ueŋ/
6	/iue/, /iuan/, /iun/, /iuŋ/
7	/i/, /iu/, /ia/, /ie/, /iai/, /iau/, /iou/, /ian/, /in/, /iaŋ/, /iŋ/

Table 4: The 38 Mandarin finals are classified into 7 final groups according to the middle vowel sound

Each Final can be divided into Medial, Kernel and Coda. There are only 3 Phonetic-like unit (PLU) can be act as medial. They are /i/ /u/ /u:/. Kernels , however, includes all the vowels. Codas has 2 vowel and 2 constant members. They are /i/ /u/ /n/ /ng/. Table 5 shows the structure of the Initials and the Finals in term of 33 PLUs. Thus, the 33 PLUs can be used to construct Initials and Finals, and hence base syllable.

Initial = [consonant]	
Consonant	/b/ /p/ /m/ /f/ /d/ /t/ /n/ /l/ /g/ /k/ /h/ /j/ /q/ /x/ /zh/ /ch/ /sh/ /r/ /z/ /c/ /s/
Final = [medial] kernel [coda]	
Medial	/i/ /u/ /u:/
Kernel vowels	/a/ /o/ /e/ /i/ /u/ /u:/ /e/ /er/
Coda (2 vowel + 2 constant)	/i/ /u/ /n/ /ng/

Table 5: The structure of Initials and Finals in term of 33 PLUs

Tonal Syllable, Base Syllable, Initial & Final, PLU are the common units of speech recognition. Table 6 summary the hierarchy of Mandarin words.

<table>
<tr><td colspan="5">Word (100,000+)</td></tr>
<tr><td colspan="5">Chinese Character (10,000)</td></tr>
<tr><td colspan="5">Tonal Syllable (1,345)</td></tr>
<tr><td colspan="4">Base Syllable (408)</td><td>Tone (5)</td></tr>
<tr><td>Initial (22)</td><td colspan="3">Final (38)</td><td>Tone (5)</td></tr>
<tr><td>Initial (22)</td><td>Medial (3)</td><td>Nucleus (9)</td><td>Ending (2)</td><td>Tone (5)</td></tr>
<tr><td colspan="4">PLU (33)</td><td>Tone (5)</td></tr>
</table>

Table 6: The hierarchy of Mandarin words, where the number inside every bracket indicates the total number of that kind of unit in Mandarin Chinese.

2.2 Acoustic Models

2.2.1 Acoustic Unit

The major goal of speech recognition is to transcribe the input speech into word strings. To accomplish this, one may wish to create word-level acoustic models for speech recognition. Nevertheless, word models are difficult to be realized directly when the vocabulary size is very large, as there may not be enough training data to train each of the words. This problem can be solved by creating sub-word models, which may be at morpheme level, syllable level, initial-final level or phoneme level. Hence, the training data can be shared across different words. The choice of units for acoustic modeling is actually one of the vital issue in speech recognition [8][9]. However, syllable and phoneme models are the most commonly used sub-word models for Chinese speech recognition. For Mandarin Chinese, there are about 408 base syllable and 34 phonemes. In this thesis, we have chosen base syllable as acoustic model for our speaker independent large vocabulary Mandarin continuous speech recognizers.

2.2.2 Hidden Markov Model (HMM)

Once the acoustic units have been chosen, we should look for a method to model it properly. There are two popular methods to model the acoustic units. They are neural network and hidden Markov model (HMM) [12][17]. Currently, the most popular method is the HMM. In fact, most exiting speech recognizers on the market use HMM to model the acoustic units of speech.

As shown in Figure 3, HMM can be view as a state machine. The states are indexed by numbers. The machine is then able to follow the arrow to change state or loop back to the current state. The state transition is actually a random process. A probability is assigned the each arrow such that each transition is base on the probability. After each transition, one output will be produced at the current state where the output set is finite. The output of a state is called an observation. The process is then referred as observation emission. If the finite states of a Markov model are not known (hidden) and only the output signal can be observed, the model is called a hidden Markov model (HMM).

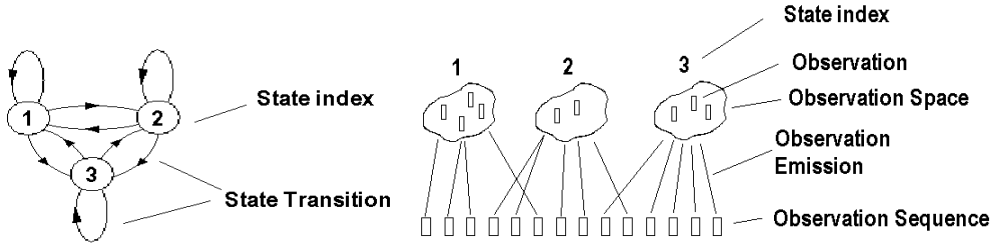

Figure 1: Illustration of state transition and observation emission.

The use of HMM for modeling acoustic units is illustrated in Figure 2. The HMM presents a word by imitating the speech production process. During the speech production process, a state is moving from left to right following the arrows. Segments of speech are generated by states in form of observation feature vector. The feature vector can be MFCC, CMS, LPC parameters [10][11][15][16], which is able to describe a speech signal. To facilitate good speech stimulation, accurate HMM parameters must be estimated which is known as training process. After training, the model is then reliable. The output of HMM will have similar acoustic features as the original signal. Once the HMM parameters are found, we can make use of the HMM to trace back which is the most likely acoustic HMM state, given an speech signal as observation. A search algorithm is then need to perform the task.

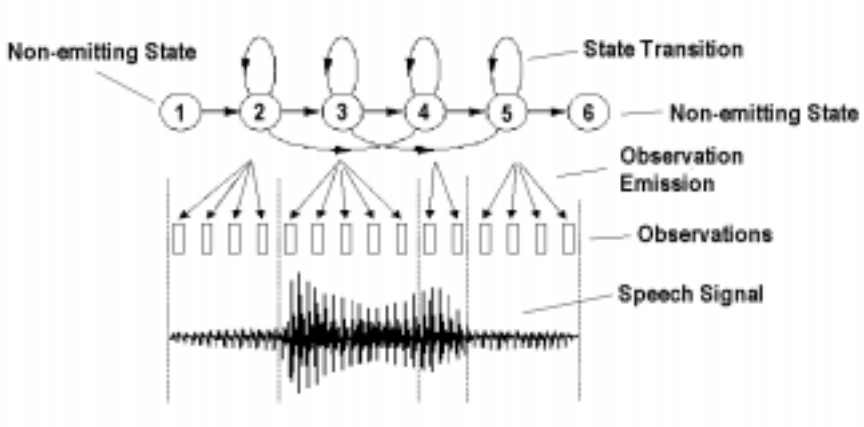

Figure 2: Illustration of HMM for modeling acoustic units

2.3 Search Algorithm

A search algorithm is employed to find the most likely HMM sequence of the acoustic unit, i.e. syllable in our case. There are two commonly used decoding algorithms. They are Viterbi decoding algorithm [18][19] and stack decoding, and we choose the Viterbi algorithm for our recognizer.

The Viterbi search is essentially a dynamic programming algorithm, consisting of traversing a network of HMM states and maintaining the best possible path score at each state in each frame. It is a time synchronous search algorithm in that it processes all states at time *t* before moving on to time *t*+1. The operation of Viterbi decoder is illustrated by the trellis diagram, which is shown in Figure 3. Each dot represents the possible HMM state at time *t*. Except from the initial state, all the dots are pointed by a arrow, which is the survive path from the previous state. The survival path is the path with the highest probability from the previous state to the current state. When the operation ended at time 6, the system would trace back which is the survival path from time 5 and so on. Such that the best state sequence is found, in this case, to be 1-1-2-2-3-4-4.

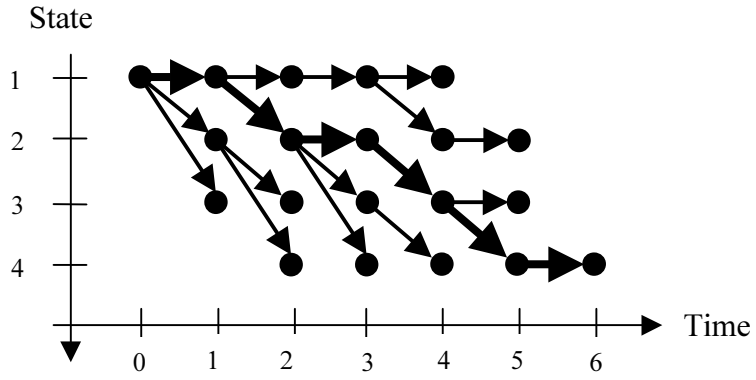

Figure 3: Illustration of Viterbi algorithm for finding the best state sequence

2.4 Statistical Language Models

The most abstract problem involved in large vocabulary speech recognition is to define an appropriate "language constraints" for the recognizer. Language constraints are generally concerned with how words may be concatenated, in what order, in what context, and in what context meaning. Language model is the most common method to realize the language constraints in a recognizer. Hence, a large-vocabulary speech recognition system is consisting of two parts: the acoustic model and the language model. The acoustic model turns the utterance into a list of candidates that are exported to the language model as its input. The language model determines the most possible word sequence.

2.4.1 Context-Independent Language Model

Context-independent language models assign probabilities to words without considering the context, i.e. history information. The simplest context-independent model of syntactic structure would simply let any word in a lexicon to be equally probable. Given the vocabulary *V* and for any word *w*, the occurring probability is:

$$P(w) = \frac{1}{|V|}$$,where $|V|$ is the size of vocabulary

This model does not have any probabilities to estimate and therefore does not need any training data. However, it is of little use to speech recognition because all words receive the same probability. It will therefore have no influence on the ranking of the words.

Another way to construct a context independent model is to estimate the probability of each word based on its relative frequency. The use of relative frequency method is also call Maximum Likelihood Estimation method. The equation of the relative frequency method is:

$$P(w) = \frac{f(w)}{\sum_{w} f(w)}$$, where *f(w)* is the frequency of word *w*

Hence, the model has only one static probability for each word. There are only *V* parameters to be estimated. Although it is a very simple language model, it is actually being used in commercial speech recognizer. Because it is a special case (*n*=1) of n-gram models, it is always referred as the unigram model.

The advantage of context-independent language model is that it requires very few training data, and uses relatively fewer parameters for the speech recognizer, which is very desirable in practical systems. However it has the inherit difficulty in predicting next word based on history information.

2.4.2 Word-Pair Language Model

The previous context-independent language models have only one distribution for each word, independent of context, while the simplest form of context dependence is a word-pair language model. The word-pair language model simply consists of a list of valid word pairs. All valid pairs are equally probable, and other pairs are impossible. Although the language model is very simple, it works very well in some tasks, such as the Resource Management task (RM). When the task has very rigid grammar, word pair models have sufficient coverage with low perplexity.

2.4.3 N-gram Language Model

If we further improve the word pair language model by adding probability to the model, so that words follow other words with differing probabilities, we get bigram model. If we then condition the probability of a word not just on the immediately preceding word, but on the preceding *n*-1 words, we get an n-gram language model. The difference between bigram, trigram and other n-gram models is just the value of *n*. The parameters of an n-gram are thus the probabilities:

$$P(w_n \mid w_1 ... w_{n-1}) \qquad \text{for all } w_1, w_2 ... w_n$$

Given a word string $S = w_1, w_2 ... w_k$, an N-gram model defines the probability of the string $P(S)$ as a product of conditional probabilities [20]:

$$P(S) = P(w_1 \mid < s >) P(w_2 \mid < s > w_1) ... P(w_m \mid < s > w_1 ... w_{-1})$$

where <s> is a special delimiter marking the start of a word string

N-gram model can be view as partitions of data into equivalence classes based on the last *n*-1 words in the history. Such that, a bigram induces a partition based on the last word in the history. A trigram model further refines this partition by considering the next-to-last word. A 4-gram model further refines the trigram, and so on.

The hierarchy of refinements introduces a tradeoff between detail and reliability. The equivalence class for bigram is the largest, such that the estimates of bigram are more reliable. While the equivalence classes of trigram are more detail but numerous, such that many of them contain only a few examples from training data, and many more are still empty. However, the differentiating power of the trigram is greater, which means that it should result in lower perplexity for the language model, given that it is well trained. Since the number of parameters in n-gram models grows exponentially with *n*, n-gram with $n>3$ is not realizable in a practical system.

The advantage of the n-gram model is that it captures the information provided by the preceding *n*-1 words. Judging from its success, this is an important source of information, especially for fixed word order language like English. Its disadvantage is the enormous amount of training data needed for obtaining all the probabilities.

2.4.4 Backoff n-gram

To model longer term dependencies, we would like *n* to be as large as possible. However, as *n* increases, the number of observation samples for each n-gram becomes less. Backoff n-gram can help to reliably estimate the probabilities. In the backoff method, the different information sources are ranked in descending order of detail or specificity. During recognition process, the most detailed model is consulted first. If it contains information about current context, it is used exclusively to generate the estimate. Otherwise, the next detailed model in line is consulted. The backoff method is simple and compact. For example, assume that there are not sufficient statistics for a particular

trigram $w_{n-1}w_{n-2}w_n$. To help us compute $P(w_n \mid w_{n-1}w_{n-2})$, we can estimate its probability by using the bigram probability $P(w_n \mid w_{n-1})$. Similarly, if we still do not have any bigram count to compute $P(w_n \mid w_{n-1})$, we can look to the unigram $P(w_n)$.

Let $w_j^k = w_j ... w_k$, the backoff n-gram model is then defined recursively as follows:

$$Pn(w_n \mid w_1^{n-1}) = \begin{cases} (1-d) \cdot c(w_1^n) / c(w_1^{n-1}) & \text{if} \quad c(w_1^n) > 0 \\ \alpha(c(w_1^{n-1})) \cdot P_{n-1}(w_n \mid w_2^{n-1}) & \text{if} \quad c(w_1^n) = 0 \end{cases}$$

where $c(w_1^n)$ is the frequency of word string w_1^n occurring in the corpus, d is the discount ratio, and α's are backoff weights.

2.5 Smoothing for Language Model

The major problem with standard n-gram models is that there are insufficient samples to train up all n-gram parameters. Thus, the resulting language model may assign a zero probability to some perfectly acceptable Chinese n-grams. This is known as zero probability problem. The task of reevaluating some of the zero-probability and low-probability n-gram, and assigning them non-zero value, is called smoothing.

Let us consider a small example, which uses a standard bigram. Let our training data S be composed of the three sentences:

(Roger read Mao Zedong Writings. Desiree read a different book. She read a book by Chris.)

To calculate *p*(*Roger read a book*). We have

$$p(Roger \mid w_{bos}) = \frac{c(w_{bos} \cdot Roger)}{c(w_{bos})} = \frac{1}{3}$$
$$p(read \mid Roger) = \frac{c(Roger \cdot read)}{c(Roger)} = \frac{1}{1}$$
$$p(a \mid read) = \frac{c(read \cdot a)}{c(read)} = \frac{2}{3}$$
$$p(book \mid a) = \frac{c(a \cdot book)}{c(a)} = \frac{1}{2}$$
$$p(w_{eos} \mid book) = \frac{c(book \cdot w_{eos})}{c(book)} = \frac{1}{2}$$

Hence,

$$p(Roger\ read\ a\ book) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

Now, consider the sentence Mao read a book. We have

$$p(read \mid Mao) = \frac{c(Mao \cdot read)}{c(Mao)} = \frac{0}{1}$$

So we have *p*(*Mao read a book*) = 0. Obviously, this is an underestimate for the probability *p*(Mao read a book) as there is some probability that the sentence occurs.

Smoothing is used to address the problem. The simplest type of smoothing technique is additive smoothing [20] which is to pretend each bigram occurs once more than it actually does.

$$p(w_i \mid w_{i-1}) = \frac{c(w_{i-1} \cdot w_i) + 1}{c(w_{i-1}) + |V|}$$

where |V| is the vocabulary size.

$$p(\textit{Roger read a book}) = \frac{2}{15} \times \frac{2}{13} \times \frac{3}{15} \times \frac{2}{14} \times \frac{2}{14} \approx 0.0001$$

$$p(\text{Mao } \textit{read a book}) = \frac{1}{15} \times \frac{1}{13} \times \frac{3}{15} \times \frac{2}{14} \times \frac{2}{14} \approx 0.00002$$

It is noticeable that the probability of "Mao *read a book* " is no longer zero, which is more sensible in practice. Actually, a more detailed analysis on smoothing can be found in chapter 5.

Chapter 3 Lexical Access

3.1 Introduction

Developing a recognizer with extremely large vocabulary size has been a challenging problem. The use of linguistic knowledge may improve the performance of the recognizer. This chapter describes a model of lexical access using partial phonetic information. Over past two decades, a variety of broad class representations for lexical access has been proposed in the literature. Yet often these proposals describe the effect of broad classes representation for western languages only.

A number of researchers have evaluated the effect of broad representation. David W. Shipman [22] investigated the statistical properties and constraints of the phonemic structures of large lexicons. Their results demonstrated broad phonetic labeling could be very useful in reducing the number of potential word candidates. For example, categorizing the sound segments in terms of six broad classes can uniquely specify about one third of the lexical entries for a 20,000-word lexicon. Daniel P Huttenlocher [23][24] described the theoretical approach to implement a large-vocabulary isolated word recognizer. The system consists of three stages. First, the classification stage produces a sequence of broad phonetic classes. Second, the sequence is used to retrieve a set of word candidates from a large lexicon. At the last stage, the subset is further extracted to identify the actual spoken word. Luciano Fissore [25] presented their large-vocabulary isolated-word recognition system which makes use of broad class pre-selection. By adding the pre-selection process to the traditional direct approach, the complexity of the new system can be reduced by 73% compared to the direct approach, while the recognition accuracy remains comparable.

These studies have provided much valuable information on the analysis, implementation and experiment results of large isolated word recognizer, which takes the advantage of broad class pre-selection process. However, broad class analysis of Chinese Language has not yet been explored. In this chapter, we will investigate the phonological and lexical characteristics of the most commonly

spoken language: Mandarin. We will compare the results with Cantonese and American English. In addition, the implementation method for a large-vocabulary isolated Chinese word recognizer will be proposed. We believe that the broad class pre-selection process will enable us to deal with the large-vocabulary recognition problem in an efficient manner. Section 3.2 presents the motivation of broad classes representation. Section 3.3 introduces the model of broad classes representation. In Section 3.4-3.5, a model of broad classes representation is presented. Section 3.6 presents the analysis of the broad classes representation. Section 3.7 proposes an implementation method that makes use of broad classes representation. Section 3.8 concludes this chapter.

3.2 Motivation: Phonological and lexical constraints

Human speech is a highly constrained system. It is known that there are various sources of constraints. 1) Production constraint: There are less than 50 phoneme in Mandarin Chinese. A person cannot speak Mandarin using another phoneme. 2) Speech recognition constraints: Different phonemes in a language tend to be distinct in perception. 3) Natural language constraints: There are syntactic, semantic and discourse level constraints for a language. We believe that constraints at the phonological and lexical levels are as important as the syntactic, semantic and discourse level.

For any language, speech is produced by a limited number of phonemes. In addition, the sequence of phonemes can only be combined in a certain way to form a meaningful word. Native speakers possess the knowledge about the word formation rules of their own language. For example, there is a set of syllable formation "rules" which governs the formation of base syllable from initials and finals [30]. Regarding the combination of an initial and a final in construction of a syllable, some restrictions are shown in Table 7. Initial /f/ cannot be followed by final starting with /i/. Therefore syllable /fing1/ is not an allowable sound in Mandarin Chinese. This information would be very useful in speech recognition. For example, initial of a syllable is either /j/, /q/ or /x/, then there is a 79% chance that the final starts with a /i/ medial. Also, there is no chance that the final starts with null medial or /u:/ medial. The example is certainly uncovering the power of phonological knowledge.

On the other hand, /de1 shi1/ is a permissible sequence of syllables in Mandarin, but is not a word because it is not in the lexicon. Hence /de1 shi1/ should not be an allowable output for a recognizer. Therefore, if we have sufficient information of what are the potential words in the lexicon, we can further constraint the sequence of syllables for a recognition task.

Hence, the phonological and lexical knowledge is presumably important in speech recognition, particularly when the acoustic cues to a speech sound are missing or distorted. Thus, we are concerned with how such knowledge can be used to constrain a speech recognition task.

	Final			
INITIAL (occur frequency)	No medial 開口	/i/ medial 齊齒	/u/ medial 合口	/u:/ medial 撮口
/b/ /p/ /m/ 5.15%,0.98%,3.74%	47.98%	33.33%	18.68%	0.00%
/f/ 2.45%	84.62%	0.00%	15.38%	0.00%
/d/ /t/ 12%, 3.53%	59.04%	20.87%	20.09%	0.00%
/n/ /l/ 2.53%, 5.69%	46.38%	41.58%	10.17%	2.03%
/z/ /c/ /s/ 3.01%,1.15%,1.08%	54.81%	0.00%	45.19%	0.00%
/zh/ /ch/ /sh/ /r/ 7.18%,2.75%,7.66%,1.94%	75.13%	0.00%	24.87%	0.00%
/j/ /q/ /x/ 6.98%,3.11%,4.86%	0.00%	78.73%	0.00%	21.27%
/g/ /k/ /h/ 5.50%,1.83%,4.42%	58.81%	0.00%	41.19%	0.00%
Φ 12.45%	5.91%	55.18	26.14%	13.59%

Table 7: The occurrence frequency of the initials and the conditional probability of the initials in combination with the finals of Mandarin. (Base on a corpus with one million syllables)

3.3 Broad Classes Representation

Many phonological rules are specified in terms of broad phonetic classes rather than specific phonemes. For example, the nasal-stop cluster rule in English specifies that nasal and stop consonant must be produced at the same place of articulation. Thus, we have words like "limp" or "can't", but not "limt" or "canp". Rather than performing detailed phonetic analysis, a word is characterized in terms of broad phonetic classes. This partial description is then used to retrieve a small set of words from a large lexicon. Our lexical study is based on Mandarin Call Home and English COMLEX obtained from the Linguistic Data Consortium. The Chinese lexicon is consisted of 44,000 words, and the English lexicon is consisted of 52,000 words.

In selecting a representation for lexical access, we try to find a classification, which can be extracted from the acoustic signal irrespective of local context, speaker characteristics, and other environmental variability. There are two common methods to classify phonemes into broad phonetics classes, that are grouping them by manner of articulation and the place of articulation. Table 8 shows the members of Mandarin consonants in each group.

Place of Articulation	Initial Consonant
Labial	/b/ /p/ /m/ /f/
Dental/Alveolar	/d/ /t/ /n/ /l/
Guttural	/g/ /k/ /h/
Palatal	/j/ /q/ /x/
Dental Sibilant	/z/ /c/ /s/
Retroflex	/zh/ /ch/ /sh/ /r/

Manner of Articulation	Initial Consonant
Stops	/b/ /p/ /d/ /t/ /g/ /k/
Laterals	/l/ /r/
Nasals	/m/ /n/
Affricates	/c/ /z/ /zh/ /ch/ /j/ /q/
Spirants	/f/ /s/ /sh/ /x/ /h/
Glides	/y/ /i/ /w/ /u/

Table 8: Place and Manner of articulation classification for Mandarin consonants

In our experiments, the six broad phonetic classes are formed based on the manner of articulation. They are "vowels, stops, fricatives, affricates, laterals/glide, nasals" in the Chinese dialects and "vowels, stops, strong fricatives, weak fricatives, laterals/glide, nasals" in the English. This set of manner classes is used, since it tends to be relatively invariant across different speakers and phonetic contexts. Zhang [26] made comparison on intelligibility of a consonant to the effect of Mandarin syllable perception.

Figure 4 shows that manner of articulation plays a more important role in identifying a syllable correctly.

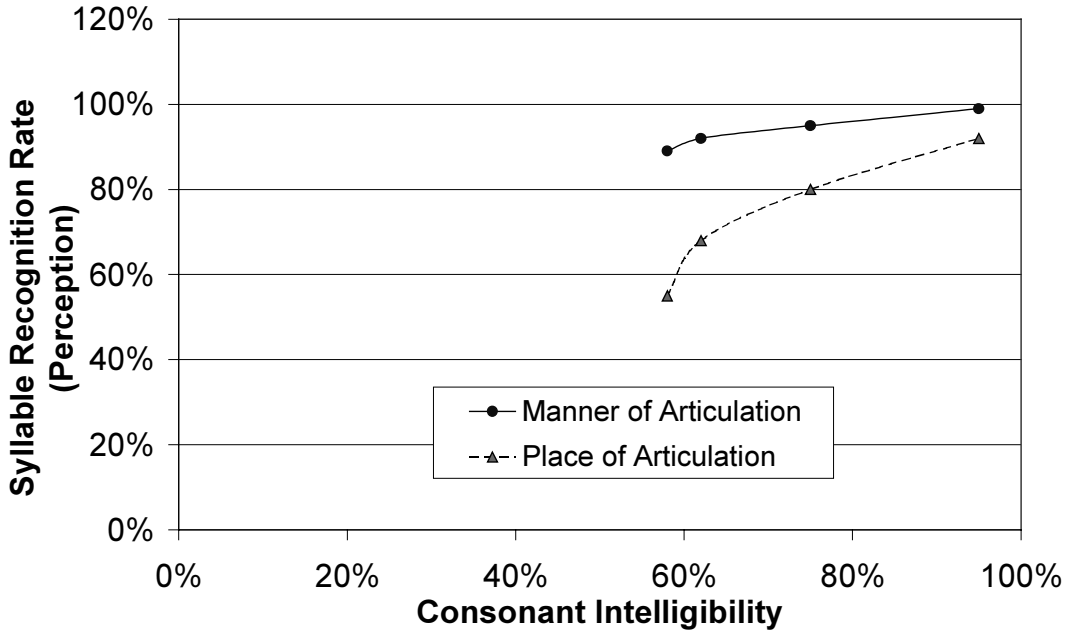

Figure 4: Manner/Place Recognition Rate vs Consonant Intelligibility [26]

In the experiment, the researcher prepared some single-syllable sound files. The subjects heard a consonant segment from the sound file and they were made to identify the consonant. The subjects then heard the whole sound file and they were made to identify the syllable. The process was performed for each prepared syllable.

Since manner of articulation consistently results in a higher syllable recognition rate than the place of articulation at the same level of intelligibility, it suggests that the manner of articulation would be more appropriate for speech recognition.

Once we identified the classifying method, we can re-label the transcription in term of the broad classes, so that it can be used for broad classes analysis and recognizer training. Broad phonetic classification can be viewed as the partitioning of the lexicon into equivalence classes of words sharing the same phonetic class pattern. For example, the characteristics of Mandarin, can be represented in terms of: 1) Tonal syllables, e.g. /nin2 men5/ 2) Base syllables, e.g. /nin men/ and 3) Manner of broad classes e.g. [Nasal] [Vowel] [Nasal] [Nasal] [Vowel] [Nasal].

For example, there are only 23 words in a 44,000-word lexicon have [Nasal][Vowel][Nasal] [Nasal][Vowel][Nasal] broad classes representation. It was found that, even at this broad phonetic level, approximately 1/5 of the words in the 44,000-word lexicon could be uniquely specified. Tonal syllables and base syllables are usual forms to represent a Chinese word. However, our experiment showed that broad class representation is also very useful in speech recognition.

3.4 Broad Classes Statistic Measures

In this section, each of the lexicons for the three different languages is represented in multiple units, including tonal syllables, base syllables, phonemes, and broad phonetic classes. In order to explore the characteristics of the languages, multiple measurements are made, such as coverage, uniqueness, expected and maximum cohort sizes. Since words in a lexicon may have very different frequency of occurrence, some of our measurements are also weighted by the frequency of occurrence. The frequency of occurrence for English is obtained from the Brown Corpus, whereas the frequency of occurrence for Mandarin and Cantonese are obtained from the Call Home database.

Table 9 shows some of the basic measurements used in our study. The maximum cohort size represents the largest equivalence class size given a particular phonetic / syllabic description, whereas the expected cohort size represents the cohort size with a frequency distribution. Notations for different measurements are shown in Table 10.

	UNIFORM DISTRIBUTION	FREQUENCY NORMALIZED
Maximum cohort size	$\max_{w_i \in L} \lvert C(w_i) \rvert$	$\max_{w_i \in L} \lvert C(w_i) \rvert$
Expected cohort size	$\frac{1}{\lvert L \rvert} \sum_{w_i \in L} \lvert C(w_i) \rvert$	$\sum_{w_i \in L} p_i \lvert C(w_i) \rvert$

Table 9: The basic measurements used in our study. $| C(w_i) |$ is the cohort size for word w_i, $|L|$ is the lexicon size, and p_i is the frequency of occurrence of the i'th word, w_i, in lexicon L.

NOTATION	STATISTICS
UNIQ	% of word which is uniquely specified
ECS	Expected cohort size
F-ECS	Frequency normalized expected cohort size
MCS	Maximum cohort size
RECS	Expected cohort size /lexicon size
F-RECS	Frequency normalized expected cohort size /lexicon size
RMCS	Maximum cohort size /lexicon size
LEX	Lexicon size

Table 10: Notations for the measurements used in this study. Results normalized by frequency of occurrence are shown in *italic*.

3.5 Broad Classes Frequency Normalization

In all languages, some words occur much more frequently than others do. The occurring probabilities of a word in the lexicon are very uneven, words like “the”, “I”, “and” occur more frequently. It would be interesting to see the frequency distribution of the words in a language. Figure 5 shows the cumulative distribution of the most frequent words for Mandarin and English. For example, the set of the most frequent 4,000 words cover over 92% and 77% of all the texts in Call Home Mandarin and the Brown Corpus, respectively.

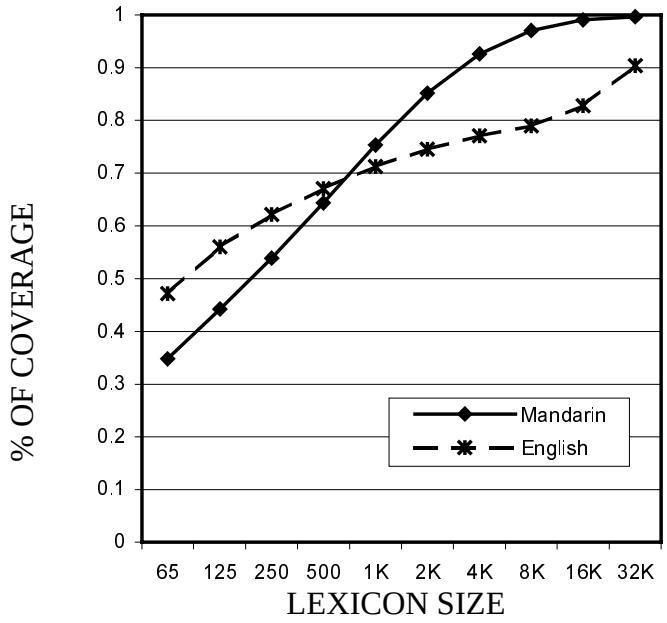

Figure 5: Percentage of text coverage for English and Mandarin most frequent words

3.6 Broad Classes Analysis

This section describes the properties of broad classes. Table 11 shows our results for Mandarin. It can be seen that if the lexicon is represented in terms of the tonal syllables, only 85% of the lexicon can be uniquely specified. The remaining 15% of the lexicon contain words that cannot be uniquely specified by the tonal syllables. This low percentage of 85% reflects the fact that many of the words in Mandarin are actually homophones. For example, all of the following Chinese words have the same tonal-syllable representation, /*fu4 shu4*/":

復述	複數	負數	富庶

When the lexicon is represented in terms of the base syllables, i.e. syllables with no tone information, only 65% of the lexicon can be uniquely specified. Similarly, only 19% of the lexicon can be uniquely specified by the broad classes.

	Tonal Syllable	Base Syllable (38 phoneme)	Manner of Articulation
UNIQ	85.0%	65.0%	19.0%
ECS	1.39	2.54	62.4
F-ECS	3.44	9.24	127.6
MCS	21	54	299
LEX	44K	44K	44K

Table 11 Analysis on Mandarin broad classes

Table 12 further describes the problem of homophone of different word length in Mandarin Chinese. The analysis was done based on a 70,687 words lexicon. It was found that 50% of total homophones are single syllable word.

Length of Words (# of syllable)	Number of Words	Number of Different Tonal-Syllable String	Number of homophones	Number of Different Base-Syllable String
1	5384	1405	3979	529
2	45602	41988	3614	31814
3	9554	9406	148	9310
4	9324	9183	141	9314
>5	823	818	5	817
Total	70687	62800	7887	51640

Table 12: Analysis of uniqueness Mandarin words in term of tonal syllable and base syllable

On the other hand, Table 11 also shows the discriminatory power of the broad phonetic classes. By using only six broad phonetic classes, the expected cohort size (ECS) is found to be 62.4. In other words, if the lexicon is represented in terms of the broad classes, on average 62.4 words would have the same broad class representation.

Table 13 shows the characteristics of Cantonese. We can see that 87%, 70%, and 16.7% of the lexicon can be uniquely specified by the tonal syllables, base syllables, and broad phonetic classes, respectively. These figures are quite similar to those for Mandarin.

However, the expected cohort size in Cantonese is 107.9, almost twice of the corresponding size in Mandarin. This shows that the broad phonological structures for the two Chinese dialects are quite different. It also suggests that the six broad phonetic classes are not as effective in differentiating the Cantonese words in the lexicon.

	Tonal Syllable	Base Syllable (38 phoneme)	Manner of Articulation
UNIQ	87.2%	70.1%	16.7%
ECS	1.32	2.15	107.9
F-ECS	2.69	6.80	165.9
MCS	26	37	471
LEX	44K	44K	44K

Table 13: Analysis on Cantonese broad classes

Table 14 shows our analysis for English. It can be seen that over 93% of the lexicon can be uniquely specified by a set of 43 phonemes, in contrast to the 85% and 87% for Mandarin and Cantonese with tonal information. Furthermore, the expected cohort size is about 74, which is comparable to the corresponding figures in Mandarin and Cantonese. These experimental results for English are very similar to those reported by Carter. We have found that the largest broad class cohort is [fricative] [Vowel] [fricative] [vowel] [fricative]. This cohort has 648 word members, such as "thesis".

	43 Phonemes	Manner of Articulation
UNIQ	93.2%	15.7%
ECS	1.07	74.1
F-ECS	1.83	111.5
MCS	5	648
LEX	52K	52K

Table 14: Analysis on English broad classes

In order to compare directly the lexical characteristics of all three languages, Table 15 summarizes the results when the base syllables (or 38 phonemes) are used for the Chinese dialects, and the set of 43 phonemes is used for English. It can be seen that the characteristics of the three languages are quite different. First, there is a major difference between the UNIQ's for the three languages, ranging from 65% for Mandarin to 93% for English. Second, the relative cohort sizes can differ by as much as a factor of 2.7, since the RECS for Mandarin is 0.0057% and the RECS for English is 0.0021%. Finally, the RMCS can also differ by an order of magnitude, since the RMCS for Mandarin is 0.12% and the RMCS for English is 0.0096%.

	Mandarin	Cantonese	English
UNIQ	65%	70.1%	93.2%
ECS	2.54	2.15	1.07
MCS	54	37	5
RECS	0.0057%	0.0049%	0.0021%
RMCS	0.12%	0.083%	0.0096%
LEX	44K	44K	52K

Table 15: Comparisons of characteristics between Mandarin, Cantonese, and English. Both Mandarin and Cantonese are based on the base syllables (or 38 phonemes), whereas English is based on a set of 43 phonemes.

We have also compared the lexical characteristics of the three languages using the 6 broad classes. Table 16 summarizes the results. We can see that their characteristics are more similar than those using the entire phoneme set. First, it is observed that almost 20% of the Mandarin lexicon can be uniquely defined by the broad phonetic classes, compared to 15.7% for English. Second, the relative expected cohort sizes are quite small for all three languages, with the highest one at 0.24% for Cantonese and the lowest one at 0.14% for both Mandarin and English. Third, while the

maximum class sizes for all three languages are still quite low, they differ by only a factor of 2. For example, the RMCS for Mandarin is 0.67%, whereas that for English is 1.25%.

	Mandarin	Cantonese	English
UNIQ	19.0%	16.7%	15.7%
ECS	62.4	107.9	74.1
MCS	299	471	648
RECS	0.14%	0.24%	0.14%
RMCS	0.67%	1.1%	1.25%
LEX	44K	44K	52K

Table 16: Analyses on Mandarin, Cantonese, and English for six broad classes

The effectiveness of the broad class representation for the three languages are compared, Figure 6 shows the relative expected cohort sizes (RECS) as functions of the lexicon sizes. It can be seen that the RECS decrease monotonically. With a lexicon size of 4,000, the RECS for all languages are below 1%.

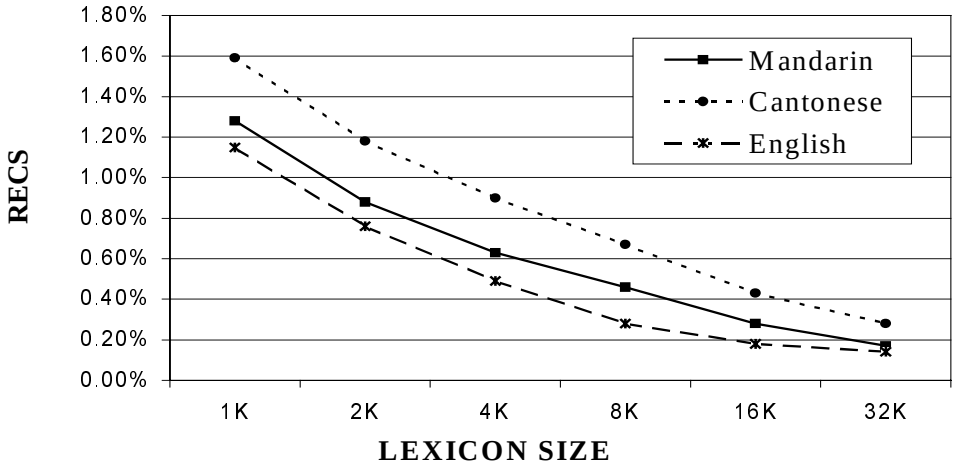

Figure 6: Relative expected cohort size analysis of 6 broad classes.

Figure 7 to Figure 10 compare the characteristics of Mandarin and English as functions of the lexicon sizes. We can see that most of the curves are quite linear with the lexicon size and that the characteristics using broad phonetic classes are quite similar between the languages.

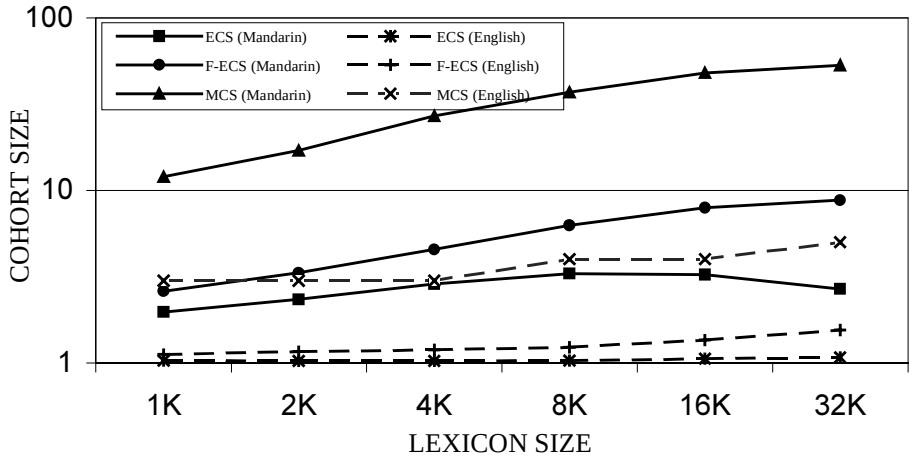

Figure 7: Expected and Maximum Cohort Size for whole phoneme set

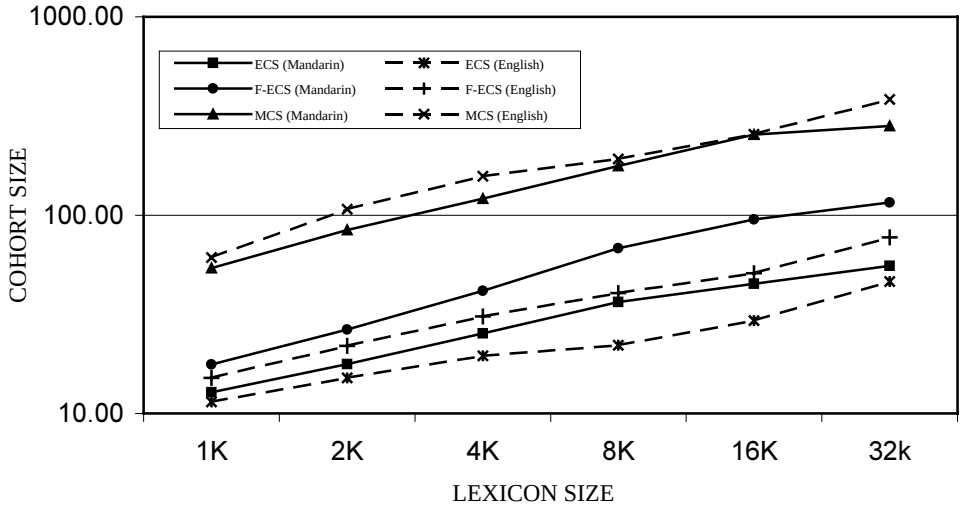

Figure 8: Expected and Maximum Cohort Size for 6 broad classes

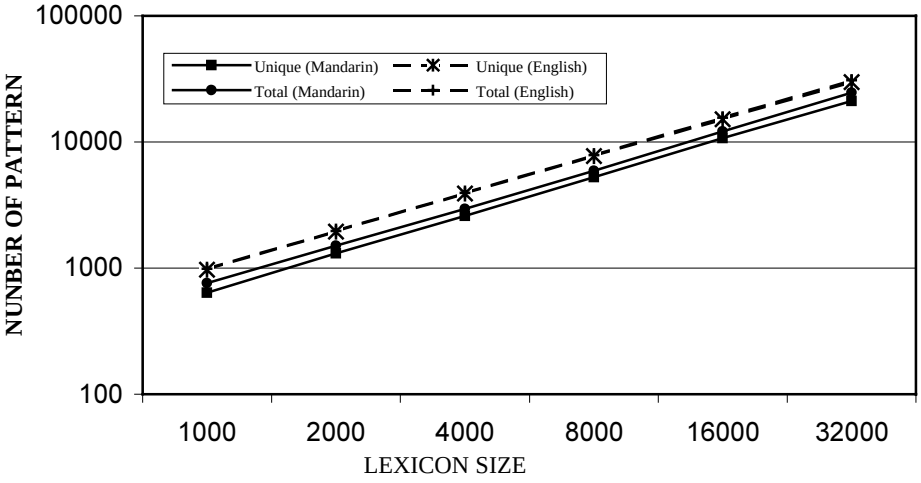

Figure 9: Total number of pattern and number of unique pattern for whole phoneme set

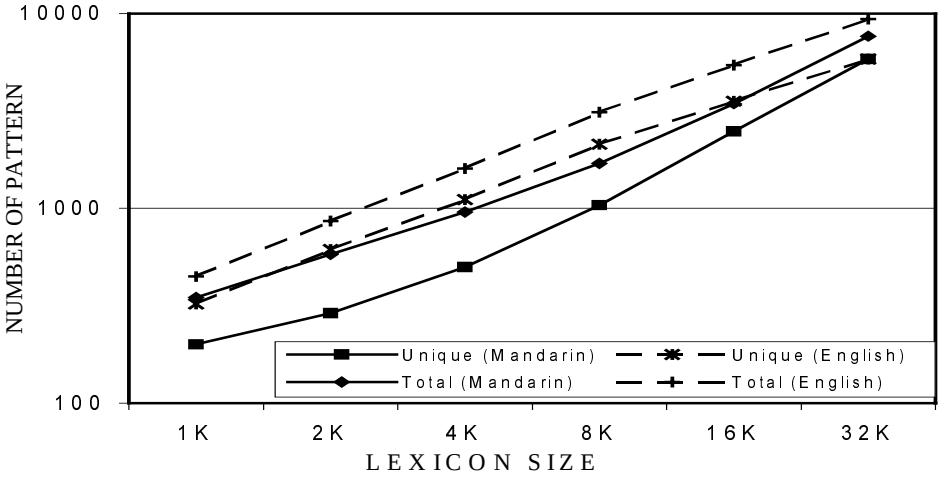

Figure 10: Total number of pattern and number of unique pattern for 6 broad classes

3.7 Isolated Word Speech Recognizer using Broad Classes

Our model of isolated word recognizer involves three distinct stages: The first stage is classification of the acoustic signal. Second, this sequence is then used to retrieve a set of word candidates form the 44,000-lexicon. Finally, a detail feature recognizer determines which of those words was actually spoken. Figure 11 shows the block bigram of the algorithm.

The recognizer has two major features. First, the classification of the speech signal is in terms of phonetic-size units as opposed to fixed rate labeling. Second, there is no attempt to perform detailed recognition of the acoustic signal until after lexical access.

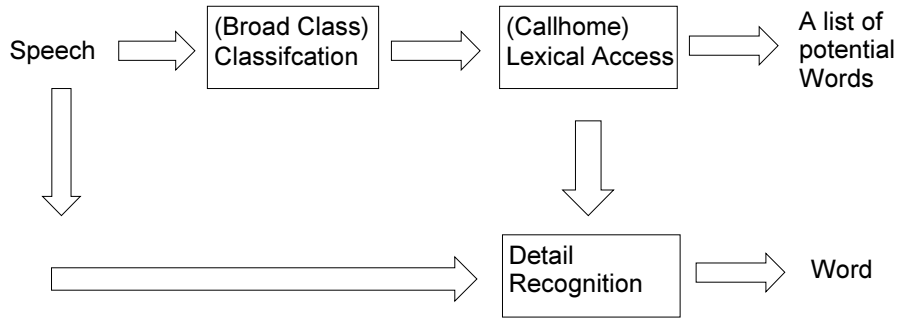

Figure 11: System Flow of the Isolated Word Speech Recognizer

Performance of the system can be measured by: 1. The number of word candidates returned in lexical access. 2. The probability of the correct word that appears in the candidate set.

By using these two separate criteria, the tradeoffs inherent in the choice of representations are more explicit. If very broad classes are extracted from the acoustic signal, then the error rate in recognizing these classes will be very low. However, a large number of words will match each sequence of the broad classes. If the classes are detailed, the error rate will be higher, but fewer words will match each sequence. We believe the 6 broad classes would be a good option for the first stage of the recognizer. Since a large vocabulary isolated word Mandarin corpus is not available in public, the mentioned system is not actually built for experimental testing.

3.8 Chapter Conclusion

In this chapter, we demonstrate a method for partitioning a large lexicon into small equivalence classes, bases on phonetic constraints. In our two-stage recognizer, word classes are represented in terms of manner of articulation. The benefits are: 1) These broad phonetic classes are relatively invariant across different speakers and phonetic contexts. 2) Only a tiny subset of the words in very large lexicon matches a given sequence of the classes.

We demonstrated that broad phonetics classification of words could, in principle, reduce the number of word candidates significantly. It is found that the Mandarin broad class representation can uniquely specific 19% words in a 44404-word lexicon, and the expected cohort size is only 62.4. Thus, a subsequence recognizer only need to search 64.2 words instead of 44404 words. It is also marked that the percentage of uniquely specified word (UNIQ) of 6 broad classes are very similar for the languages, they are 19.0%, 16.7% and 15.7% for Mandarin, Cantonese and English respectively. Since the broad class recognizer only makes use of the broad features of an input phoneme, minor change in the acoustic realization would not affect the result of first lexical access stage. Thus, the representation is both powerful at differentiating between words, and robust with respect to acoustic variability.

Therefore, lexical access, through the broad classes feature, is undoubtedly a feasible way to cut down the computation time of a large-vocabulary isolated-word recognizer.

Chapter 4 Character and Word Language Model

4.1 Introduction

N-gram language model simultaneously encodes syntax, semantics and pragmatics. They concentrate on local dependencies. It is especially effective for structural languages, such as English where word order is important and the contextual effects among neighbor words is strong. On the other hand, n-gram language model processes inherit deficiencies in exploiting long-range constraints. Researchers have tried different approaches to solve the problem [31][32][33]. However, these attempts have yielded little improvement at the high expense of computational cost. Thus, in this thesis, we concentrate on n-gram language model only.

People tend to speak more freely (less constraint in syntax or grammar) in telephone conversation. It makes the building of language model even more difficult. Not much work of language model has been done for Chinese large-vocabulary telephone speech. IBM [39] has presented their experiments of Call Home corpus. However, the detailed analysis of the language model has not been published. In this chapter, we analyze and compare the characteristics of Chinese word language model and character language model.

While word language model has been found to provide powerful constraints for speech recognition, it is also known that word language model suffers from out-of-vocabulary and sparse data problems. These problems are particularly severe in Chinese, as new Chinese words can be created with a high degree of freedom. Furthermore, as there are no clearly defined word boundaries in Chinese, some forms of word segmentation procedures must first be performed before word language model can be applied. In this chapter, we explore the possibility of using character language model, which can potentially alleviate some of the known problems with word language model.

4.2 Motivation

While many researchers assume that word level language model is better than character level language model. Few researchers compared the performance of both language models. The motivation of the chapter comes from the fact that the assumption may not be true, and it is worth to have systematic procedures for the comparison. In our experiments, the performances of language models are compared based on the perplexity.

Perplexity

The most common response after experiencing large-vocabulary speech recognizer is "It doesn't make sense!". The better the language model we have, the lower the occurrence of nonsense sentences. How can we identify a better language model? Language model is commonly measured by "perplexity" which is the extent of constraints of a given language model in a recognizer. This term roughly means the average number of branches at any decision point during the decoding of the message. For a simple language model, in which all of the *V* words is allowed to follow any word with probability $1/V$. The perplexity of this model is V. This concept can be extended further, where the probability of words following each other is not uniformly $1/V$. From Rabiner & Juang [34], perplexity *B*, is defined in term of entropy *H*. $B = 2^H$

And we estimate *H* over Q words of data to be H_p:

$$H_p = -\frac{1}{Q}\log P(w_1, w_2, \ldots w_q)$$

Which for an n-gram model is:

$$H_p = -\frac{1}{Q}\sum_{i=1}^{Q}\log P(w_i \mid w_{i-2}, \ldots w_{i-N+1})$$

In practice, we can only estimate probabilities using some test data, and thus only an estimation of perplexity can be obtained. The more data is used to train and test the language model, the better this estimate should be.

For speech recognition, fewer possible words means an easier task for the recognizer. Hence, a language model with low perplexity is more desirable. It will generally result in faster and more accurate recognition. The relationship between perplexity and word accuracy is not guaranteed, although we expect models with lower perplexity introduce better recognition results.

4.3 Call Home Mandarin corpus

4.3.1 Acoustic Data

Our study is based on the second release (apr95) of Mandarin Call Home corpus [37][38], distributed by Linguistic Data Consortium (LDC). The CallHome Mandarin corpus consists of 120 telephone conversations between native speakers of Mandarin. All speakers were aware that they were being recorded. They were given no guidelines concerning what they should talk about. Once a caller was recruited to participate, he/she was given a free choice of whom to call. Most participants called family members or close friends overseas. All calls originated in North America. The corpus is a large-vocabulary, conversational and telephone speech corpus. Speech is transcribed and time-aligned with human intervention. Conversations take place in an unprompted manner with no specified topics for talker to follow. Each recording is 10 to 30 minutes long. The transcription is in native orthography, covering 10 minutes of each call. Unlike Switchboard [40], transcription in Call Home is time-aligned interactively by speaker turns instead of on a word-by-word basis.

Because of the Call Home corpus speech is collected over international connections, there are channel noise and distortions to deal with. Moreover, handsets and speakerphones are often used in the case of multiple talkers on one end. The average number of speakers per conversation in Mandarin Call Home is 2.81 instead of 2 for Switchboard. In view of poor quality of telephone speech, not all the transcribed speech from conversation is suitable for training. After the non-Mandarin speech, laughter and corruptive channel noise are removed, the usable portion of training speech is about 9.0 hours.

There are 80 conversations in the training set and 20 in the development test set. The training set contains 19K sentences and 5744 unique words. The average word length is about 1.39 characters. Detailed statistics of the corpus can be found in Table 17.

Transcription	Training	Development Testing
# of dialog	80	20
# of sentence	19,965	5,378
# of word	127,063	34,699
# of character	177,148	48,218
# of unique word (w-vocab.)	5,774	2,936
# of unique character(c-vocab.)	2,098	1,466
Average word length	1.394 character/word	1.390 character/word

Table 17: Detailed statistics of Call Home Corpus

4.3.2 Transcription Texts

Mandarin Call Home exhibits strong characteristics of spontaneous speech with lots of disfluencies, hesitations, repetitions of phrases, and word slurring, which makes human transcription complicated. Unlike Switchboard corpus, no conversation topic is specified in Call Home. Talkers speak in a more relaxing manner in Call Home than in Switchboard because they are family members or close friends. Moreover, proper names such as human names and abbreviations for organizations frequently appear in the context.

The speech is transcribed in native orthography, Chinese characters, by human transcribers. There are several problems for Mandarin transcription, which do not occur in western languages. For example, word boundaries in Mandarin are ambiguous and cannot be clearly distinguished by simple rules [41]. Some examples of texts used as test sentences are as follows.

<s> 剛 來 那時候 感覺 特別 不好 </s>
<s> 他 說 有 什麼 事兒 可以 找 他 </s>

To have better analysis of my testing result, the characteristic of usable portion of training speech data and transcription of IBM's experiment [39] and those of my experiments are compared in Table 18. It is notice that IBM include less transcription for training (7.7hrs vs. 9.8hrs). It might account for the small variation of our results.

Corpus	IBM-Call Home	CUHK-Call Home
# of Recording(Training/ Devtest)	80/20	80/20
Transcribed length of training	7.7 hrs	9.8 hrs
# of Turns	12,000/3,000	19,965/5,378
# of Words for LM	170K	177K
Trigram Perplexity	288	313.23

Table 18: Baseline perplexity compare to IBM

4.4 Methodology: Building Language Model

A bigram language model can be seen as a *V*x*V* matrix of probabilities, where *V* is the size of the vocabulary in a specific task. The bigram and trigram probabilities can be estimated by the simple relative frequency approach.

$$Bigram: P(w_n \mid w_{n-1}) = f(w_n \mid w_{n-1}) = \frac{c(w_{n-1}, w_n)}{c(w_{n-1})}$$

$$Trigram: P(w_n \mid w_{n-2}, w_{n-1}) = f(w_n \mid w_{n-2}, w_{n-1}) = \frac{c(w_{n-2}, w_{n-1}, w_n)}{c(w_{n-2}, w_{n-1})}$$

where the function *c*(.) counts the number of string in the blanket

The use of relative frequencies as a way to estimate probabilities is known as Maximum Likelihood Estimation (MLE) [35][36]. Table 19 and Table 20 show the bigram and unigram counts of 7 words in Call Home corpus. The relative frequencies are then calculated by normalizing the bigram with their unigram counts. Table 21 shows the bigram probabilities after normalization. Note that 'N/A' log probabilities are caused by zero bigram counts, which is undesirable. Actually, we have chosen 7 sample words which are more related to each other, the majority count of full-version bigram matrix should be zero. From the Table 19, we also notice that the disfluency problem is very severe, for example, there are 259 number of bigram 我-我, 73 number of bigram 他-他 and 17 number of bigram 什麼-什麼. Generally, those bigrams would not be appeared in written text. However, people tend to repeat their words in telephone conversation.

W_{n-1} \ W_n	我	想	知道	他	買	了	什麼
我	259	126	119	6	10	6	4
想	21	2	1	7	6	5	1
知道	50	0	12	13	0	30	4
他	11	6	5	73	10	17	7
買	0	1	0	0	4	50	3
了	100	0	2	44	6	1	7
什麼	7	0	1	5	0	12	17

Table 19: Bigram count for 7 words (out of 5774) in Call Home Corpus

Unigram	我	想	知道	他	買	了	什麼
Count	4568	423	571	2246	281	2953	881

Table 20: Unigram count for 7 word in Call Home Corpus

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	-1.1929	-1.5058	-1.5306	-2.8280	-2.6062	-2.8280	-3.0041
想	-1.2576	-2.2788	-2.5798	-1.7347	-1.8016	-1.8808	-2.5798
知道	-0.9657	N/A	-1.5855	-1.5507	N/A	-1.1875	-2.0626
他	-2.2843	-2.5476	-2.6268	-1.4624	-2.3257	-2.0953	-2.4806
買	N/A	-2.3483	N/A	N/A	-1.7462	-0.6493	-1.8712
了	-1.4461	N/A	-3.1450	-1.8026	-2.6679	-3.4461	-2.6010
什麼	-2.0787	N/A	-2.9238	-2.2248	N/A	-1.8446	-1.6933

Table 21: Log-probabilities of bigrams for 7 word in Call Home Corpus

As illustrated above (N/A entries in log-probabilities table), a bigram/trigram language model would give zero probability to string *W* which contains an unseen type of bigram/trigram. If the bigram/trigram missing-rate is high, a large number of word errors would be introduced in a recognizer, which operates with the statistical decision criterion:

$$P(\hat{W})P(A \mid \hat{W}) = \max_{W} P(W)P(A \mid W)$$

Actually, we found that 58% of the word trigrams and 26% of the word bigrams appearing in the test set never took place in the training set. One approach to solve the zero probability problem is using backoff technique. In the model, the probability backoff from a trigram to a bigram, and then to a unigram estimation. The ideal is incorporated in the approximated formula.

$$P(w_i \mid w_{i-2}, w_{i-1}) = \begin{cases} \widetilde{P}(w_i \mid w_{i-2}, w_{i-1}) & \text{if} \quad c(w_{i-2}, w_{i-1}, w_i) > 0 \\ \alpha(w_{n-2}^{n-1})\widetilde{P}(w_i \mid w_{i-1}) & \text{if} \quad c(w_{i-2}, w_{i-1}, w_i) = 0 \quad \text{and} \quad c(w_{i-1}, w_i) > 0 \\ \alpha(w_{n-1})\widetilde{P}(w_i) & \text{otherwise} \end{cases}$$

where $\alpha(w_{n-2}^{n-1}), \alpha(w_{n-1})$ are factors that depend on the counts *c* and assure that the probability *P* when summed over all words w_i adds up to 1.

$\widetilde{P}$ is a discounted version of bigram and trigram. The discount method used for the simple backoff bigram in our experiments is shown below:

$$Bigram: \widetilde{P}(w_n \mid w_{n-1}) = \left(1 - \frac{1}{c(w_{n-1})}\right) \frac{c(w_{n-1}, w_n)}{c(w_{n-1})}$$

An intuitive impression for the quality of the language model can be conceived from Table 22. The simple backoff language model is used to predict the potential words for the sentence "<s> 他 都 不 知道 他 的 條件 有 多麼 好 </s>". Table 22 manifests all the words that are predicted to be more likely than the actual word, given the language model has perfect knowledge of the preceding word. For example, knowing the preceding word " 有", the language model estimates that the most likely next word is "一個", and the word 機會,時間… are all more likely than the actual word "多麼" which is estimated as the 219-th likeliness, given that particular past.

We observe that the language model is quite effective at predicting most function words (e.g. 我,你,他) but that is uncertain about some content words (e.g. 條件, 多麼). Another observation is that the language model provides powerful constraints to a speech recognizer. In the above example, the correct words are always within the top 800 candidates, instead of 5774 words. It is quite amazing that without any acoustic information of the current word, a recognizer is able to predict accurately the potential words. Thus the searching time for the recognizer can be greatly reduced. Of course, we cannot make the conclusion merely based on the example. An objective and quantity measure of language model quality would be presented in the following sections.

Rank	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}
1	<s>	嗯	說	是	知道	</s>	說	</s>	</s>	一個	大	</s>
2		呃	</s>	不		嗎	</s>	時候	了	有	是吧	
3		哦	他			我	他	那個	她	什麼	</s>	
4		哎	也			吧	也	啊	也	</s>	我	
5		啊	就			了	就	我	也好	個	你	
6		我	是			啊	是	你	我	可能	啊	
7		對	現在			你	現在	東西	啊	這個	的	
8		你	的			他	的	啦	的	機會	了	
9		那	那個					是	不好	一	呃	
10		他	要					人	大家	一些	嗯	
11			在					信	都	了	他	
12			不					嘛	對吧	那個	就	
13			還					他	好	空調	哦	
14			這個					呀	好啊	兩	那	
15			那					那	啦	我	對	
16			去					事情	吶	好	哎	
17			都					這	你	時間	那個	
…								…	…	…	…	
34								一個	說	都	奸	
…								…	…	…		
40								錢	有	傳真		
								…		…		
219								或者		多麼		
...								…				
782								體質				
783								天				
784								天天				
785								條件				

Table 22 Word prediction by a backoff bigram language model for "<s> 他 都 不 知道 他 的 條件 有 多麼 好 </s>"

4.5 Character Level Language Model

Out-of-vocabulary (OOV) rate and perplexity of different character language model are analyzed. There are 156 out-of-vocabulary characters in the test set, some of them with higher OOV count are listed in Table 23. The total number of OOV word token is 298, which represents 0.62% of the total number of words in the test set.

Count	11	10	8	8	7	7	7	7	7	5	5	5	5	4	4	4	4	4	3	3
OOV	肺	勹	徽	需	弘	贾	椒	蚊	靴	槐	盔	膜	疫	迪	岡	搖	耀	忠	扑	蔥
Count	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
OOV	甘	龜	杭	狠	牧	禹	織	醃	辦	玻	醇	轟	宏	漸	框	璃	乃	膩	瞧	釋
Count	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
OOV	汰	淘	恬	誘	裕	蚤	柠	逐	煮	尊	罔	呱	呶	泓	梓	痱	T	挨	礙	叭

Table 23: List of top frequency out-of-vocabulary characters

Table 24 exhibits the perplexity and OOV rate of the simple backoff character bigram and trigram. Figure 12 visualizes those perplexity results. Eight language models are compared in Table 24. They are divided into two groups: cheating and fair models. Cheating models are included in the experiments to give a wider range of perplexity analysis. Since bigram and trigram language models are the most common types of language model implementation methods, both of them are employed in the experiment. Smoothing techniques are also included, aimed at further improving the perplexity of language model. More detailed analysis of smoothed technique would be presented in Chapter 5.

Cheating language models are language models, which are trained from testing transcription. Hence the cheating language models has better statistic information than fair language models, which trained on training transcription. The cheating language models always have lower perplexity than the fair ones, and would normally perform better in recognition tasks. Each group contains four members. They are bigram/trigram language model with/without smoothing. Since many possible Chinese character trigrams w_1,w_2,w_3 never actually take place even in very large corpora of training

text, it is noticeable that the simple backoff character trigram (Perplexity, PP=96.03) has a higher perplexity than the simple backoff character bigram (PP=63.13). It suggests that there is insufficient training data for trigram language model. Hence, most of the subsequence experiments are worked with bigram language model.

It is also noticeable that smoothing technique reduces the perplexity of fair language models, but it increases the perplexity of cheating language models. It suggests that the smoothing technique would not be usefully if the target language model already has good statistic information.

Furthurmore, it is found that OOV rate of character language model is only 0.62%, while it is 4.15% (shown in Table 26) for word language model. For a recognizer without using statistical language model, each of the 2098 character is equally probable to follow any word. As described in section 4.2, its perplexity is 2098. By using the simple backoff language model, the perplexity is reduced from 2098 to 63.13, i.e. 97% improvement.

Character Level Language Model	**Test on**	**Perplexity/ Entropy (Character)**	**OOV OOV (%)**
Fair Bigram (no smoothing)	Testing transcription Hit on 2-gram:50,663 (86.34%) Hit on 1-gram:8,013 (13.66%)	63.13 / 5.98 bits	298 hits 0.62%
Fair Trigram (no smoothing)	Testing transcription Hit on 3-gram: 34720(59.17%) Hit on 2-gram: 15943(27.17%) Hit on 1-gram: 8013(13.66%)	96.03 / 6.59 bits	298 hits 0.62%
Fair Bigram (GT smoothing)	Testing transcription Hit on 2-gram:50,663 (86.34%) Hit on 1-gram:8,013 (13.66%)	44.10 / 5.47 bits	298 hits 0.62%
Fair Trigram (GT smoothing)	Testing transcription Hit on 3-gram: 34720(59.17%) Hit on 2-gram: 15943(27.17%) Hit on 1-gram: 8013(13.66%)	43.06 / 5.43 bits	298 hits 0.62%
Cheating Bigram (GT smoothing)	Testing transcription	21.70 / 4.48 bits	0 hits 0%
Cheating Trigram (GT smoothing)	Testing transcription	11.20 / 3.48 bits	0 hits 0%
Cheating Bigram (no smoothing)	Testing transcription	17.19 / 4.10 bits	0 hits 0%
Cheating Trigram (no smoothing)	Testing transcription	5.90 / 2.56 bits	0 hits 0%

Table 24: Character level Language Model for CALL HOME spoken speech transcription

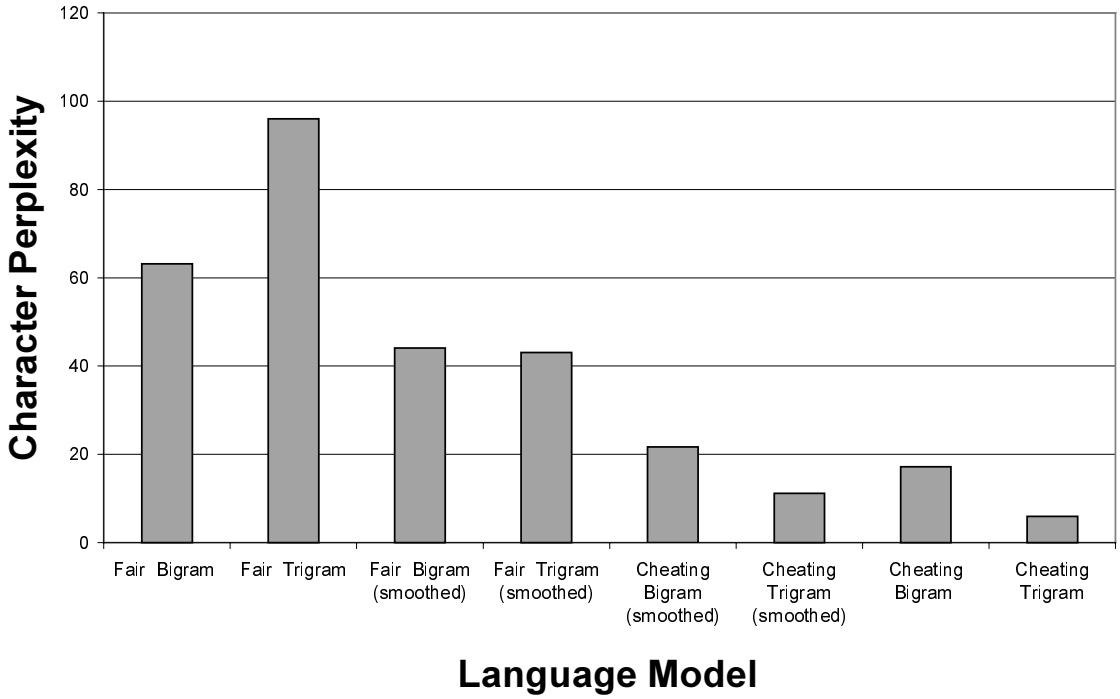

Figure 12: Character Perplexity Character level language model

4.6 Word Level Language Model

The advantage of using word language model is that it provides a better description of the language. However, it suffers from large out-of-vocabulary (OOV) rate. There were 845 OOV words in the test set. Some of them with higher OOV count are listed in Table 25. Most of these words occurred only once. The total number of out-of-vocabulary word token is 1441, which represents 4.15% of the total number of words in the test set. We notice that most of the OOV for the character level language model are rarely used character, while the OOV for the word level language contains many frequently used word, such as 這是,西瓜.

Count	18	16	13	11	10	9	9	9	9	8
OOV	真的	這是	高溫	好多	勹	季虹	酒精	糖	甜甜	生活費
Count	7	7	7	7	7	7	7	7	7	6
OOV	肺部	花椒	化療	徽	賈玫	夢超	西瓜	霞	圓	吃素

Table 25: List of top frequency out-of-vocabulary words

Moreover, Chinese has no clearly defined word boundaries. Thus, some forms of word segmentation procedures must be performed before the word language model can be applied. In addition, the Chinese lexicon contains more than 40,000 words, and therefore there are potentially $40{,}000^n$ n-grams, making sparse data a challenging problem.

An perplexity and OOV analysis, which is similar to the analysis of character language model (in section 4.5), is done for word language models. Table 26 and Figure 13 exhibit the perplexity of the word language models. Similar to character language model, the improvement in perplexity is very substantial for word level language model. The word perplexity reduced from 5776 to 175.13, which is also 97% improvement. Moreover, the simple backoff word trigram (PP=313) has higher perplexity than the simple backoff word bigram (PP=175). By comparing to the Table 11 and Table 12, it is found that both character and word language gives similar effects when similar changes are applied. Such as changing from bigram to trigram or applying smoothing technique on unsmoothed language models.

Word Level Language Model	Test on	Perplexity/ Entropy (WORD)	OOV OOV (%)
Fair Bigram (no smoothing)	Testing transcription Hit on 2-gram:32615 (74.10%) Hit on 1-gram:11399 (25.90%)	175.13 / 7.45 bits	1441 hits 4.15%
Fair Trigram (no smoothing)	Testing transcription Hit on 3-gram:18,654 (42.38%) Hit on 2-gram:13,961 (31.72%) Hit on 1-gram:11,399 (25.90%)	313.23 / 8.29 bits	1441 hits 4.15%
Fair Bigram (GT smoothing)	Testing transcription Hit on 2-gram:32615 (74.10%) Hit on 1-gram:11399 (25.90%)	90.88 / 6.51 bits	1441 hits 4.15%
Fair Trigram (GT smoothing)	Testing transcription Hit on 3-gram:18,654 (42.38%) Hit on 2-gram:13,961 (31.72%) Hit on 1-gram:11,399 (25.90%)	94.51 / 6.56 bits	1441 hits 4.15%
Cheating Bigram (GT smoothing)	Testing transcription	29.91 / 4.90 bits	0 hits 0%
Cheating Trigram (GT smoothing)	Testing transcription	15.47 / 3.95 bits	0 hits 0%
Cheating Bigram (no smoothing)	Testing transcription	18.35 / 4.20 bits	0 hits 0%
Cheating Trigram (linear smoothing)	Testing transcription	4.96 / 2.31 bits	0 hits 0%

Table 26: Word Level Language Model for Call Home spoken speech transcription

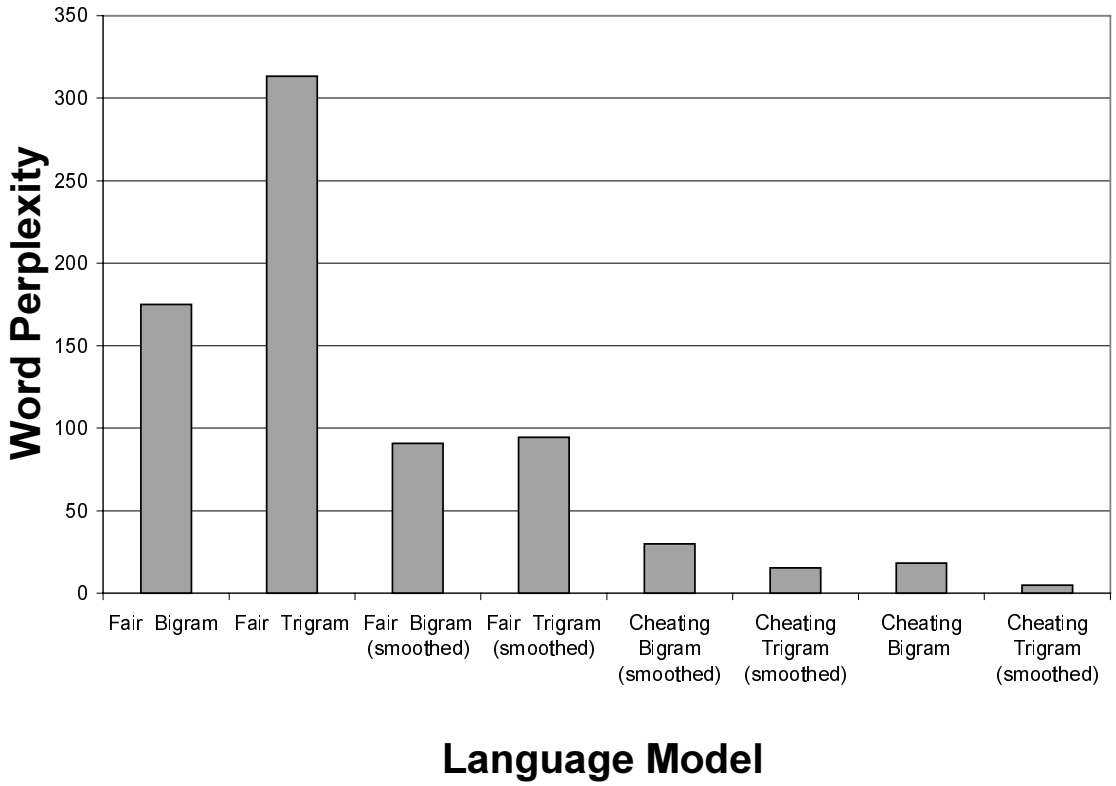

Figure 13: Word Perplexity of Word level Language Model

4.7 Comparison of Character level and Word level Language Model

The perplexities of language models presented in Table 24 and Table 26 are on the two different bases. Perplexity in the Table 24 is based on a word-level analysis, whereas perplexity in Table 26 is based on a character-level analysis. In order to compare information on the two bases, we define $PP_C = \sqrt[L]{PP_W}$.

where PP_W is the average word perplexity, PP_C is the average character perplexity, and L is the average length of a word .

Let the entropy of a sentence is *E*. Sentence entropy is equal to the sum of character or word entropy, ie. $E = \sum Ec = \sum Ew$. Since the average word length is *L* character per word, the *L* times the average character entropy ($\overline{E}c$) would equal to average word entropy ($\overline{E}w$). Hence, $\overline{E}w = L \cdot \overline{E}c$. By translating the entropy formula to perplexity formula, we provide $PP_C = \sqrt{PP}$.

After normalizing the perplexities to the same basis, we found that word language model has a better performance on perplexity, while character language model has a better performance on OOV rate.

Hence, we need to adjust OOV rate of both language models, such that the two language models are comparable. The easiest method to adjust the OOV rate is to change the vocabulary size. Table 27 and Table 28 describe the effect of vocabulary size on the two language models. Figure 14 and Figure 15 present those data in graphical form. For both language model, we found that increasing the vocabulary size reduces the OOV rate. However, increasing the vocabulary size also increase the perplexity of the language model, which is not desirable. In other words, increasing the vocabulary size of a speech recognition system has two conflicting effects. 1) reduces the OOV rate, which reduces OOV related recognition errors; 2) the added lexical entries increase the average acoustic confusability of words, which results in recognition errors. Hence, compromise must be made on certain vocabulary size, such that the problems of large OOV rate and high

perplexity are minimized. In this thesis, extensive recognition experiments have not been done to found the optimal vocabulary size. Nevertheless, OOV rate and perplexity of language models with different vocabulary sizes are compared.

Vocab. Size	PP* (Word)	Entropy	PP (character)	OOV	OOV (%)	2-gram hit	1-gram hit
2100	*193.09*	5.46 bits	44.10	298	0.62%	86.34%	13.66%
2000	*192.06*	5.46 bits	43.93	333	0.69%	86.58%	13.42%
1750	*187.16*	5.43 bits	43.12	464	0.96%	86.93%	13.07%
1500	*180.67*	5.39 bits	42.04	668	1.39%	87.44%	12.56%
1250	*172.95*	5.35 bits	40.74	946	1.96%	88.10%	11.90%
1000	*163.24*	5.29 bits	39.08	1400	2.90%	89.08%	10.92%
750	*151.40*	5.21 bits	37.02	2135	4.43%	90.40%	9.60%
500	*131.22*	5.06 bits	33.40	3759	7.80%	92.64%	7.36%
250	*104.69*	4.83 bits	28.39	7828	16.23%	95.17%	4.83%

Table 27: Effect of vocabulary size on Character Level Language Model

Vocab. Size	PP (Word)	Entropy	PP* (character)	OOV	OOV%	2-gram hit	1-gram hit
5774	90.88	6.51 bits	*25.64*	1441	4.15 %	74.10%	25.90%
5500	89.88	6.49 bits	*25.44*	1481	4.27%	76.13%	23.87%
5000	88.47	6.47 bits	*25.15*	1586	4.57%	76.76%	23.24%
4500	87.11	6.44 bits	*24.87*	1665	4.80%	77.20%	22.80%
4000	85.40	6.42 bits	*24.52*	1772	5.11%	77.61%	22.39%
3500	82.66	6.37 bits	*23.95*	1947	5.61%	78.24%	21.76%
3000	80.52	6.33 bits	*23.50*	2102	6.06%	78.86%	21.14%
2500	76.82	6.26 bits	*22.72*	2382	6.86%	79.86%	20.14%
2000	73.02	6.19 bits	*21.91*	2702	7.79%	80.95%	19.05%
1500	67.46	6.08 bits	*20.69*	3239	9.33%	82.61%	17.39%
1000	60.15	5.91 bits	*19.06*	4049	11.67%	84.99%	15.01%
500	46.48	5.54 bits	*15.83*	6141	17.70%	89.83%	10.17%

Table 28: Effect of vocabulary size on Word Level Language Model

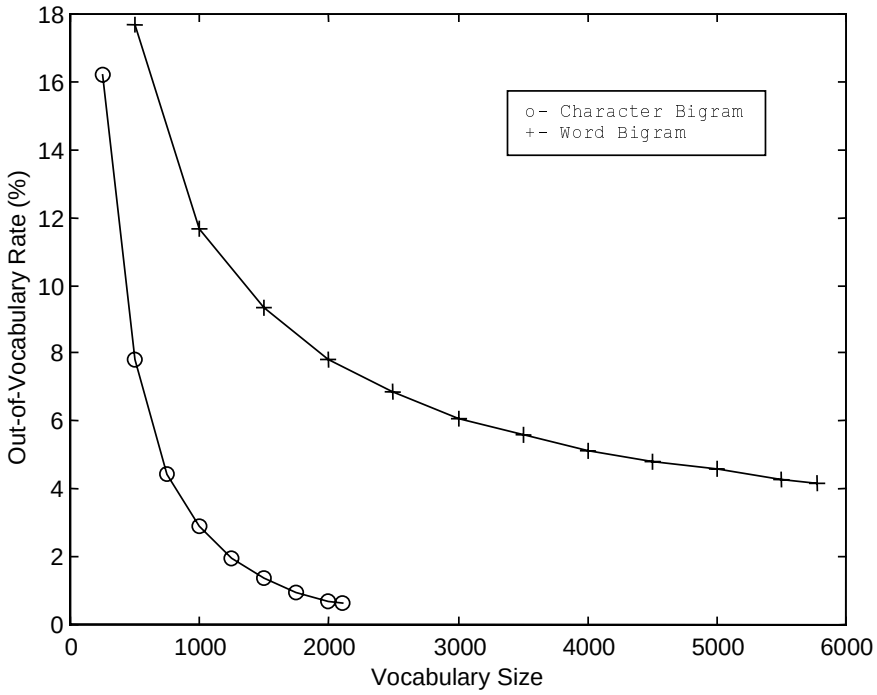

Figure 14: Effect of vocabulary size on Perplexity of Word & Character Level Language Model

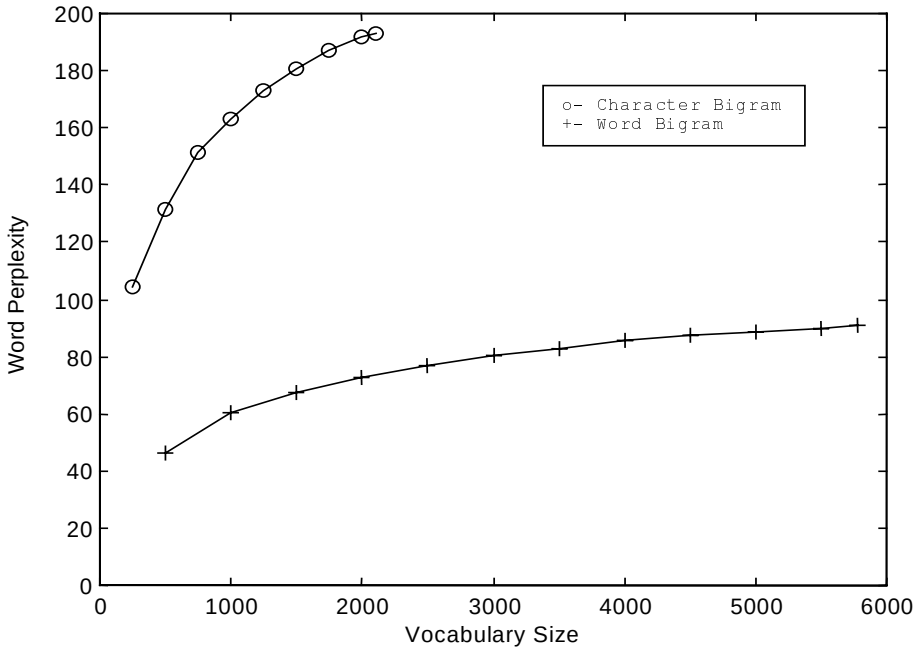

Figure 15: Effect of vocabulary size on OOV% Word & Character Level Language Model

Figure 16 compares the word perplexity of the word-level and character-level language models, as function of the OOV. As usual, we notice that increasing the vocabulary size always reduces the OOV rate. We also found that the word-level bigram consistently results in lower word perplexity than the character-level bigram, suggesting that the word-level language model maybe more appropriate for speech recognition.

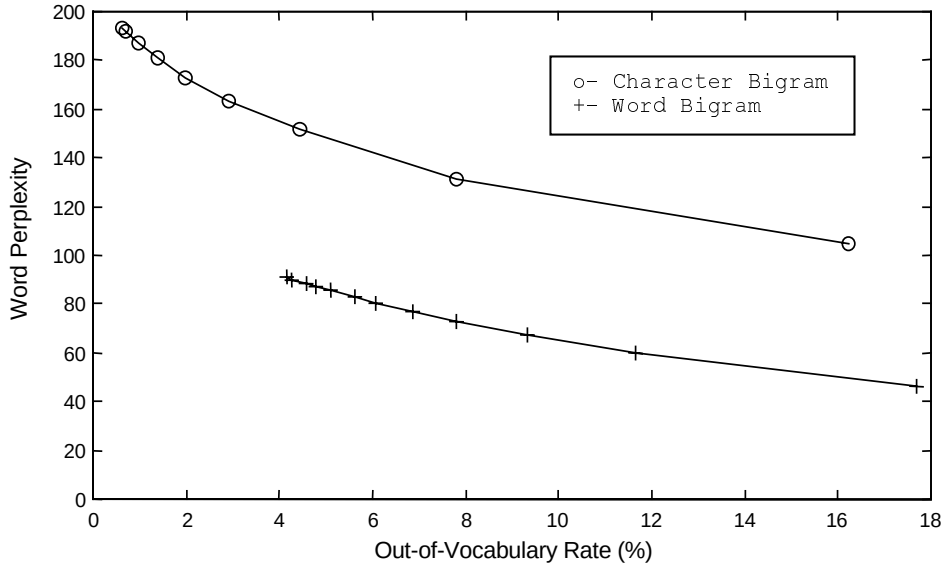

Figure 16: Perplexity vs. OOV % for Word Language Model & Character Language Model

4.8 Interpolated Language Model

We are estimating a language model for the specific domain task, Call Home speech recognition, where the amount of training data is limited. There is actually a large amount of data available from another domain, such as newspaper. A language model trained on the larger corpus may be more robust, but will not match the target domain. A simple solution, which take advantages of the large corpus language model and the specific domain language model, is to interpolate the parameters of both language models. In this experiment, we make use of another text corpus from LDC, which is HUB5 corpus. A HUB5 language model is generated base on the corpus.

4.8.1 Methodology

The probability of a given sentence assigned by the interpolated model is defined as a weighted sum of the probability assigned by the original models:

$$P(w \mid ILM) = (1-\alpha)P(w \mid CHLM) + \alpha P(w \mid HBLM)$$

$$P(w \mid ILM) = \begin{cases} (1-\alpha)P_{CH\_2}(w_n \mid w_{n-1}) + \alpha P_{HB\_2}(w_n \mid w_{n-1}) \\ (1-\alpha)P_{CH\_2}(w_n \mid w_{n-1}) + \alpha\beta_{HB\_1}(w_{n-1})P_{HB\_1}(w_n) \\ (1-\alpha)\beta_{CH\_1}(w_{n-1})P_{CH\_1}(w_n) + \alpha P_{HB\_2}(w_n \mid w_{n-1}) \\ (1-\alpha)\beta_{CH\_1}(w_{n-1})P_{CH\_1}(w_n) + \alpha\beta_{HB\_1}(w_{n-1})P_{HB\_1}(w_n) \end{cases}$$

where α is the interpolation ratio, β is the backoff weight, and CH-LM stand for Call Home LM, HB-LM stand for HUB5 LM.

The weight α is found by using the estimation maximization (EM) algorithm [43], which minimizes the perplexity of the interpolated model over the training data. The Interpolated Language Model Bigram is then generated through the formula above. There are four possible cases, i.e. both CH-LM and HB-LM has the bigram, only CH-LM has the bigram, only HB-LM has the bigram, both CH-LM and HB-LM do not have the bigram.

4.8.2 Experiment Results

Both perplexity and OOV rate can be improved by interpolating the Call Home language model with HUB5 conversation transcription. By using the Estimation Maximization (EM) algorithm, it is found that the perplexity is optimized when α=0.2.

Figure 17 shows the change of word perplexity at different interpolation ratio for simple backoff language models. There is 6.3% improvement in perplexity.

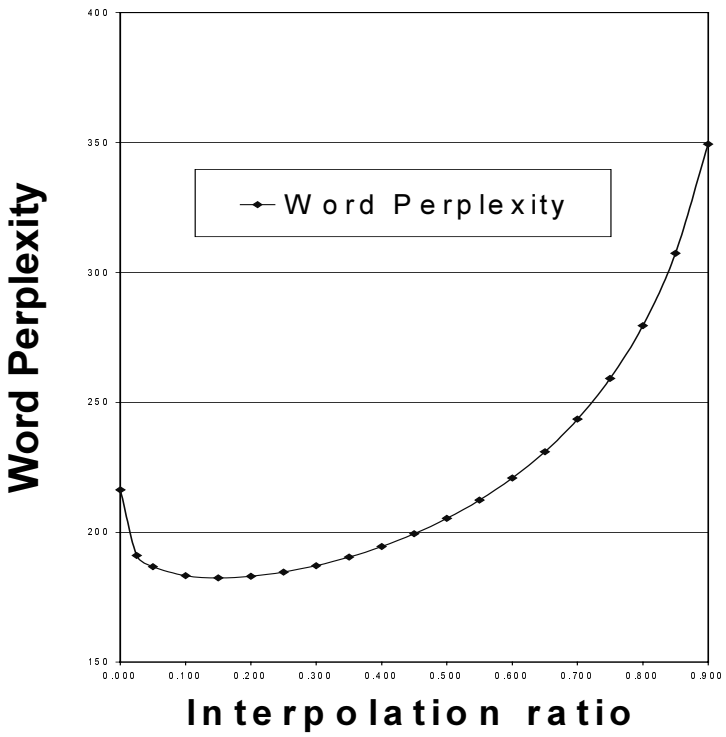

Figure 17: Word Perplexity of (Call Home & HUB5) Interpolated Language Model

The improvement of perplexity and OOV rate is less significant in the smoothed language model but it is still noticeable. As shown in Table 29, our interpolation methods have successfully reduced the OOV by 6.8%. In addition, the perplexity improves from 90.88 to 88.90.

Language Model	Perplexity	OOV
Smoothed FWB	90.88	1441
Interpolated and Smoothed FWB	88.90	1343

Table 29: Perplexity and OOV reduction of Interpolated Language Model

4.9 Chapter Conclusion

In this chapter, we demonstrate a method for implementing language model for a large vocabulary speech recognizer. We compare the performance of language models by an objective measure, perplexity. The simple backoff character bigram reduces the perplexity from 1468 to 63.13, whereas the simple backoff word bigram reduces the perplexity from 5776 to 175.3. In other words, both bigram language models reduce the perplexity by 97%. This shows that language model provide powerful constraints to a recognizer. By comparing the language models at wide range of OOV and at same perplexity unit, it shows that word level language model always gives a lower perplexity. It suggests that word level language model is more appropriate for speech recognition. We also demonstrate an interpolation method, which further reduce the perplexity of language model. By using the interpolated language model, we achieved 6.8% reduction in OOV and 2.17% reduction in perplexity. Furthermore, some perplexity results for smoothed language mode are also presented, and detailed discussion on smoothing techniques can be found in Chapter 5.

Chapter 5 N-gram Smoothing

5.1 Introduction

The problems discussed in the previous chapter are related to the estimation of $P(w_n|w_{n-2},w_{n-1})$ for trigram or $P(w_n|w_{n-1})$ for bigram based on the relative frequency $f(w_n|w_{n-2},w_{n-1})$ and $f(w_n|w_{n-1})$. Although backoff technique helps to solve the zero probability problem, the backoff model is incapable to estimate the actual probability of unseen events well. We have found that there is a severe sparse data problem for both language models. For example in language modeling for a 6000-word vocabulary, there are 36 million possible word bigram. Nevertheless, for a specific task, the training corpus rarely has more than 2 million words. A direct approach to improve the language model is to derive it from a much larger training corpus. However, this approach introduces other problems. First, we may not able to get a sufficiently large corpus (e.g. hundred million words). Second, the resulting language model would still be confounded to a specific domain, from which it was extracted. They are the two major motivations for researchers to work in a better smoothing method.

To overcome the drawbacks of the conventional maximum likelihood estimation and the incapability of the simple backoff n-gram, a number of different approaches have been proposed. Such as floor method [44], discounting technique related to the Good-Turing formula [45][46], Witten-Bell discounting [47], linear and absolute discounting [48]. However detailed comparison for the effectiveness of those smoothing methods on Chinese text has not been done.

In this chapter, we describe our work on the smoothing of n-gram models. Four smoothing techniques, which significantly improve the existing trigram models, are explained in detail. We also present and extensive empirical comparison of the smoothing techniques, which was previously lacking in the literature.

5.2 Motivation

In this chapter, we describe and compare different smoothing techniques. Smoothing is one of the most important technique to solve the zero probability problem of language models. Many smoothing techniques are proposed by researchers. The motivation of performing series of experiments in this chapter is to identify the most appropriate smoothing method for Call Home Mandarin speech, and perhaps the result can be extended to other Mandarin telephone speech. Four faovous smoothing technique, which is developed for western language, are borrowed in the experiments. They are Witten-Bell smoothing, Good-Turing smoothing, linear and absolute smoothing.

5.3 Mathematical Representation

Any language model can be seen as a probability generator. It predicts the probability of next word base on the statistics in the training data and the preceding words of testing data [49]. However, the training data is always limited, so that it fails to observe some typical event classes. Let us denote the event class under consideration by $k = 1,2,...,K$. Their sample count is then denoted as c_k, which means the number of event *k* observed in the training text. The corresponding probability for event *k* is then denoted as *p*(*k*). In other words, we then have *C* independent trials with *K* possible outcomes, where the sample counts N_k denote the number of trial resulting in outcome k.

The maximum likelihood estimation for *p*(*k*) is $p(k) = \frac{c_k}{C}$. However, most of the events *k* are never seen in the training text because there are many more event classes *K* than the number of observed event C. Thus most of the event classes has zero outcoming probability, due to $C_k = 0$. The problem of sparseness of data can be captured by the following equation: $c_1 < C << c_0 < K$.

Since all event classes k with the same sample count *c* must be assigned with the same probability and are therefore grouped into the same equivalence class. Therefore, we define n_r as the number of class numbers. The value n_r is then referred as count frequency because it is the number of classes that was observed exactly *r* times. Except add-one smoothing, each of the smoothing technique described in throughout this chapter makes use of the value n_r. Table 30 summarizes the frequently used notation in this chapter.

Symbol	Meaning
V	Vocabulary Size
$c(w_{n-1}, w_n)$	Bigram count for w_{n-1} followed by w_n
$c(w_{n-1})$	Count for unigram w_{n-1} Count for bigram with prefix word w_{n-1}
c_i	Original counts for n-gram *i*
c_i^*	Discounted counts for n-gram *i*
d_i	Discount ratio = c_i^* / c_i
g_i	Different between original count and discounted count: $g_i = c_i^* - c_i$
K	Total number of n-gram event
C	Total count of n-gram event For unigram N = V ; for bigram N = related unigram count
n_c	Number of n-gram that occur exactly *c* times
R	Number of observed event
R(*w*)	Number of observed bigram with prefix *w*
Z	Number of unseen event
Z(*w*)	Number of unseen bigram with prefix *w*

Table 30: Frequently used notation for smoothing techniques

5.4 Methodology: Smoothing techniques

Each of the mentioned smoothing methods assign a non-zero probability to the unseen events by discounting each count $c(k) > 0$, then redistributing the discounted probability mass over all n_0 unseen n-gram. The first one to be introduced is the add-one smoothing method. Add-one smoothing is also called floor-smoothing method, which give each n-gram a floor count 1. Add-one smoothing is very easy to implement, but it is a poor method of smoothing. The weakness of add-one is that it is worse at predicting the unseen n-gram probabilities. Gale and Church summarize a number of problems with the add-one method [50]. The next mentioned smoothing method is Witten-Bell smoothing method. It is only slightly more complex than add-one smoothing but gives much better results. The main idea behind the Witten-Bell smoothing is that "Use the count of things you have seen to help estimating the count of things you haven't seen." This idea gives a simple but useful estimation for unseen events. A more complex smoothing method call Good-Turing smoothing method is also introduced. The basic idea of Good-Turing smoothing is to re-estimate the low or zero count n-gram by looking at the number of n-grams with a higher count. The last two smoothing methods are absolute and linear smoothing. They are sharing the same idea that each original count is subtracted by certain value. In the case of absolute discount, the subtracting value is an 'absolute' constant. Therefore, we have 'absolute discount' as her name. Simultaneously, the subtracting value of linear discount is 'linear' to the original count. Hence, we have the name 'linear discount'. To give a rough estimation of the complexity of each method, Table 31 manifests the number of lines of program code for each of the implementation.

Smoothing method	Lines of C coding
Add-one	40
Witten-Bell	250
Good Turing	300
Absolute	150
Linear	150

Table 31: Implementation complexity of different smoothing methods

5.4.1 Add-one Smoothing

A simple way to perform smoothing is add-one smoothing. It is by just taking the matrix of n-gram counts and adding one to all the counts. Although this algorithm rarely used in smoothing of language model, it introduces important concept that would be used in other smoothing methods. An example of add-one smoothing is shown below:

Add-one smoothed bigram probability are computed by normalizing each row of modified counts by recalculating unigram count:

$$\widetilde{P}(w_n \mid w_{n-1}) = \frac{c(w_{n-1}, w_n) + 1}{c(w_{n-1}) + V}$$

where c(w_{n-1}, w_n) is the bigram count for w_{n-1} followed by w_n, and c(w_{n-1}) is the unigram count for word w_{n-1}.

The add-one smoothed language model can also be described in terms of discount ratio (d_c). It is the ratio of discounted counts (c*) to the original counts (c):

$$d_c = \frac{c^*}{c} \qquad where \quad c^*(w_{n-1}, w_n) = (c(w_{n-1}, w_n) + 1)\frac{c(w_{n-1})}{c(w_{n-1}) + V}$$

Table 19 shows the add-one smoothed counts for the bigram mentioned in chapter 4. Table 21 shows the log-probabilities of add-one smoothed bigram. You may notice that the simply add-one smoothed solves the zero probability problem in original bigram. For example, the original language model would give zero probability to a string "什麼 想 買", since the P(想 | 什麼) equal to zero. It would introduce recognition error if the string were really appeared in the testing sentence. With add-one smooth technique, P(想 | 什麼) is now equal to 3.35E-04 instead of zero.

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	260	127	120	7	11	7	5
想	22	3	2	8	7	6	2
知道	51	1	13	14	1	31	5
他	12	7	6	74	11	18	8
買	1	2	1	1	5	51	4
了	101	1	3	45	7	2	8
什麼	8	1	2	6	1	13	18

Table 32: Add-one smoothed bigram counts for 7 words in Call Home Corpus

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	-1.4603	-1.7715	-1.7961	-3.0302	-2.8339	-3.0302	-3.1763
想	-2.1838	-3.0491	-3.2252	-2.6231	-2.6811	-2.7481	-3.2252
知道	-1.8374	-3.5449	-2.4310	-2.3988	-3.5449	-2.0536	-2.8460
他	-2.6353	-2.8694	-2.9363	-1.8453	-2.6731	-2.4592	-2.8114
買	-3.5075	-3.2064	-3.5075	-3.5075	-2.8085	-1.7999	-2.9054
了	-1.7657	-3.7700	-3.2929	-2.1168	-2.9249	-3.4690	-2.8670
什麼	-2.6786	-3.5817	-3.2807	-2.8036	-3.5817	-2.4678	-2.3264

Table 33: Log-probabilities of add-one smoothed bigram for 7 word in Call Home Corpus

The effect of add-one smoothing can be manifested by reconstructing the count matrix from the probability in Table 34. Each probability in Table 34 is multiplied by its original unigram count, so that the number of smoothed unigram count is then preserved as original. Note that add-one smoothing has made a very big change to the counts: c(我 想) changed from 259 to 158.27. The effect can also be seen in the log-probability table: P(想 | 我) decrease from -1.5058 (Table 21) in the un-smoothed bigram to -1.7715 in the add-one smoothed bigram. The big change in the counts and probabilities is due to large portion of probability mass is moved to the unseen events.

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	158.27	77.31	73.05	4.26	6.70	4.26	3.04
想	2.77	0.38	0.25	1.01	0.88	0.76	0.25
知道	8.30	0.16	2.12	2.28	0.16	5.05	0.81
他	5.20	3.03	2.60	32.07	4.77	7.80	3.47
買	0.09	0.17	0.09	0.09	0.44	4.45	0.35
了	50.65	0.50	1.50	22.56	3.51	1.00	4.01
什麼	1.85	0.23	0.46	1.38	0.23	3.00	4.15

Table 34: Add-one smoothed bigram counts (reconstructed) for 7 words in Call Home Corpus

5.4.2 Witten-Bell Discounting

Witten-Bell discounting [47] is based on a Poisson process formulation for the appearance of new tokens, which was originally introduced to estimate the number of unseen biological species [46]. Later it was applied to estimate the rate of appearance of new words in natural language text [51].

Witten-Bell can be viewed as a generalized Laplace's of succession, which only deal with binary event. Supposing that there are *k* of event types instead of the two. Out of complete set of *C* observations, it supposes there are c_1 of type 1, c_2 of type 2 and so on. These random variables are related by: $c_1 + c_2 + ... + c_k = C$.

Witten-Bell estimates the total probability mass of all the unseen types with the number of observed types divided by the number of tokens *C* plus observed types R:

$$\text{P[next token will be of the i}^{\text{th}}\text{ type]} = \frac{c_i}{C + R}$$

$$\text{P[next event will be novel]} = \frac{R}{C + R}$$

We can apply the Witten-Bell formula to our smoothing problem. For example, the probability of an unseen bigram $w_{n-1}w_{n-2}$ is calculated by using the probability of seeing a new bigram starting with w_{n-1}. Note that the number of seen bigram types *R* and the number of bigram token *C* are conditioned by the previous word w_{n-1}.

$$p^*(w_i \mid w_{i-1}) = \frac{R(w_{i-1})}{Z(w_{i-1})[c(w_{i-1}) + R(w_{i-1})]}$$

where Z is the total number of bigram with zero count. Each of the formerly zero bigram now gets its equal share of redistributed probability mass. For the non-zero count bigram, we then discount them through the same manner:

$$p^*(w_i \mid w_{i-1}) = \frac{c(w_{i-1}w_i)}{c(w_{i-1}) + R(w_{i-1})}$$

To calculate the Witten-Bell discount for 7 word Call Home experiments, we need the number of bigram $R(w_{n-1})$ which has been seen in the training text. The values for the selected words in the Call Home corpus are listed in Table 35.

Bigram with Prefix	我	想	知道	他	買	了	什麼
# of Bigram	617	184	122	505	95	465	312

Table 35: The number of seen bigram for 7 words in Call Home Corpus

Since the vocabulary size is *V*, *V*=5776, and there are exactly *V* potential bigrams which can begin with a given word *w*, the number of unseen bigram is $Z(w) = V - R(w)$.

Bigram with Prefix	我	想	知道	他	買	了	什麼
# of Bigram	5159	5592	5654	5271	5681	5311	5464

Table 36: The number of unseen bigram for 7 words in Call Home Corpus

Table 37 shows the re-estimated probability of Witten Bell smoothing method. For comparison reason, the bigram count of Witten Bell smoothing method is re-constructed on Table 38.

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	-1.301	-1.614	-1.639	-2.937	-2.715	-2.937	-3.113
想	-1.461	-2.482	-2.783	-1.938	-2.005	-2.084	-2.783
知道	-1.142	-4.507	-1.762	-1.727	-4.507	-1.364	-2.239
他	-2.398	-2.661	-2.741	-1.576	-2.439	-2.209	-2.594
買	-4.352	-2.575	-4.352	-4.352	-1.973	-0.876	-2.098
了	-1.534	-4.591	-3.233	-1.890	-2.756	-3.534	-2.689
什麼	-2.232	-4.320	-3.077	-2.378	-4.320	-1.997	-1.846

Table 37: Witten Bell smoothed log-probabilities for 7 word in Call Home Corpus

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	228.180	111.006	104.839	5.286	8.810	5.286	3.524
想	14.634	1.394	0.697	4.878	4.181	3.484	0.697
知道	41.198	0.018	9.887	10.711	0.018	24.719	3.296
他	8.981	4.899	4.082	59.599	8.164	13.879	5.715
買	0.012	0.747	0.012	0.012	2.989	37.367	2.242
了	86.396	0.076	1.728	38.014	5.184	0.864	6.048
什麼	5.169	0.042	0.738	3.692	0.042	8.862	12.554

Table 38: Witten Bell smoothed bigram count for 7 words in Call Home Corpus

5.4.3 Good Turing Discounting

The Good-Turing (GT) discount method is suggested by Turing and developed by Good [53]. For certain n-gram occurs *c* times, Good-Turing re-estimated it to *c** with the formula:

$$c^* = (c+1)\frac{n_{c+1}}{n_c}$$

N_c is the number of n-gram that occur exactly *c* times. Hence, N_0 is the number of bigrams with zero count, and N_1 is the number of bigram, which occurred only once. For example, the modified count for unseen bigrams is then estimated by dividing the number of singleton by the number of unseen bigrams. Table 39 gives an example of the use of Good-Turing discount to the bigrams for the Call Home task. The first column is the original count *c*. The second column is the number of bigram which has the related count *c*. Thus, 2677 bigrams has a count of 3. The third column shows the Katz Good-Turing smoothed count.

c	n_c	*c**(GT)
0	33308413	0.0011686
1	38923	0.3141021
2	6925	1.1052781
3	2677	1.9248795
4	1332	2.8544832
5	779	4.1867221
6	550	4.8287431
7	385	6.0956292

Table 39: Good Turing smoothed bigram counts for words in Call Home Corpus

Since we assume the large bigram/trigram counts are reliable, the discounted *c** is not used for all counts *c* in our experiment. We adopt the Katz [45] modified Good-Turing model for our experiments, in which only counts less than eight ($1 \le c \le 7$) are modified.

Since the total number of smoothed count must not be changed after smoothing, a remedied equation is developed to preserve the total count.

Since the counts given to unseen n-gram are: $c_0 = n_0 * (0+1)\frac{n_1}{n_0} = n_1$,

and in order to preserve the total count, we must have $\sum_1^k n_r(c_r - c_r*) = n_1$

The unique solution to the above equation is:

$$c* = c\frac{\frac{(c+1)}{c}\frac{n_{c+1}}{n_c} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}$$

Table 40 shows the smoothed count of 7 words in Call Home Corpus by Katz Good-Turing method.

The log probability is shown in Table 41.

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	259.000	126.000	119.000	4.829	10.000	4.829	2.8545
想	21.000	1.105	0.314	6.096	4.829	4.187	0.3141
知道	50.000	0.001	12.000	13.000	0.001	30.000	2.8545
他	11.000	4.829	4.187	73.000	10.000	17.000	6.0956
買	0.001	0.314	0.001	0.001	2.854	50.000	1.9249
了	100.000	0.001	1.105	44.000	4.829	0.314	6.0956
什麼	6.096	0.001	0.314	4.187	0.001	12.000	17.0000

Table 40: Good Turing smoothed bigram counts for 7 words in Call Home Corpus

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	-1.246	-1.559	-1.584	-2.976	-2.660	-2.976	-3.204
想	-1.304	-2.583	-3.129	-1.841	-1.943	-2.004	-3.129
知道	-1.058	-5.689	-1.677	-1.643	-5.689	-1.280	-2.301
他	-2.310	-2.668	-2.730	-1.488	-2.351	-2.121	-2.566
買	-5.381	-2.952	-5.381	-5.381	-1.993	-0.750	-2.164
了	-1.470	-6.403	-3.427	-1.827	-2.786	-3.973	-2.685
什麼	-2.160	-5.877	-3.448	-2.323	-5.877	-1.866	-1.715

Table 41: Good Turing smoothed Log Probability for 7 words in Call Home Corpus

5.4.4 Absolute and Linear Discounting

The probability *P*r for absolute discounting and linear discount [48][56] can then be written in a general forms:

$$P_r = \frac{1}{C}(c_r - g_r) \quad \text{if } c_r > 0 \qquad\qquad P_r = \frac{1}{C} \cdot \frac{1}{n_0} \sum_{s>0} n_s g_s \quad \text{if } c_r = 0$$

where g_r is defined as the differences between the original count *c* and the smoothed count *c**

For absolute discount, $g_r = b$ where *b* is a constant. Intuitively it is equivalent to sample subtracting the constant b from each count. For linear discounting, $g_r = \alpha c_r$ where α is a constant. In other word, the original count is subtracted by a value, which is in proportion to its original count. Unlike the Good-Turing method, the discounting function is applied to all non-zero count.

Absolute discounting define the probability as,

$$P_r = \frac{c_r - b}{C} \quad \text{if } c_r > 0 \qquad\qquad P_r = b \cdot \frac{R}{CZ} \quad \text{if } c_r = 0$$

where *b* is a constant with value: $b = \frac{n_1}{n_1 + 2n_2}$.

Linear discount is then defining the probability as,

$$P_r = (1 - \alpha)\frac{c_r}{C} \qquad \text{if } c_r > 0 \qquad\qquad P_r = \frac{\alpha}{Z} \quad \text{if } c_r = 0$$

where α is a constant with value: $\alpha = \frac{n_1}{C}$.

Table 42 and Table 43 show the re-estimated probability by absolute and linear smoothing methods. Although the absolute and linear smoothing are relatively easy to be calculated, their perplexity performance in our Call Home experiments are as effective as the Good-Turing and Witten Bell smoothing methods.

In our Call Home task, *b* value for bigram is: $b = \frac{38923}{38923 + 2 \cdot 6925} = 0.73756$, while α value for bigram is: $\alpha = \frac{38923}{167002} = 0.23307$

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	258.262	125.262	118.262	5.262	9.262	5.262	3.262
想	20.262	1.262	0.262	6.262	5.262	4.262	0.262
知道	49.262	0.016	11.262	12.262	0.016	29.262	3.262
他	10.262	5.262	4.262	72.262	9.262	16.262	6.262
買	0.012	0.262	0.012	0.012	3.262	49.262	2.262
了	99.262	0.065	1.262	43.262	5.262	0.262	6.262
什麼	6.262	0.042	0.262	4.262	0.042	11.262	16.262

Table 42: Absolute smoothed bigram counts for 7 word in Call Home Corpus

$W_{n-1} \backslash W_n$	我	想	知道	他	買	了	什麼
我	198.635	96.633	91.265	4.602	7.669	4.602	3.068
想	16.106	1.534	0.767	5.369	4.602	3.835	0.767
知道	38.347	0.077	9.203	9.970	0.077	23.008	3.068
他	8.436	4.602	3.835	55.986	7.669	13.038	5.369
買	0.038	0.767	0.038	0.038	3.068	38.347	2.301
了	76.693	0.426	1.534	33.745	4.602	0.767	5.369
什麼	5.369	0.124	0.767	3.835	0.124	9.203	13.038

Table 43: Linear smoothed Bigram counts for 7 word in Call Home Corpus

The different between two discounting models is that absolute discount affects the high counts much less than low counts, while linear discounting scales down all counts by the discount factor $d_r = (1 - \alpha)$.

5.5 Comparison of Different Discount Methods

We have explored the effectiveness of the four different discount methods to overcome the sparse data problem. These discount methods are: linear, Witten Bell, absolute, and Good Turing. As Figure 18 shows, compare to the simple backoff method, all the four smoothing methods give more than 39% reduction in perplexity for the Call Home task. Good-Turing smoothing is the most effective one among those methods. Specifically, the Good-Turing smoothed word bigram and character bigram perplexities are 90.88 and 193.09 respectively. These results compare favorably to the simple backoff bigram perplexities of 175.13 and 317.93. As shown in Table 44, it is quite amazing that the simple absolute discount method gives the perplexity value 91.97, which is better than the more well-known and more sophisticated Witten-Bell algorithm.

Discount Method	Word-LM Perplexity	Character-LM Perplexity
Simple backoff	175.13	$63.13^{1.39} = 317.93$
Good Turing	90.88	$44.10^{1.39} = 193.09$
Absolute	91.97	$44.25^{1.39} = 194.01$
Witten Bell	93.88	$44.47^{1.39} = 195.35$
Linear	106.00	$48.22^{1.39} = 218.62$

Table 44: Comparison of different Discounting Methods for the Call Home task

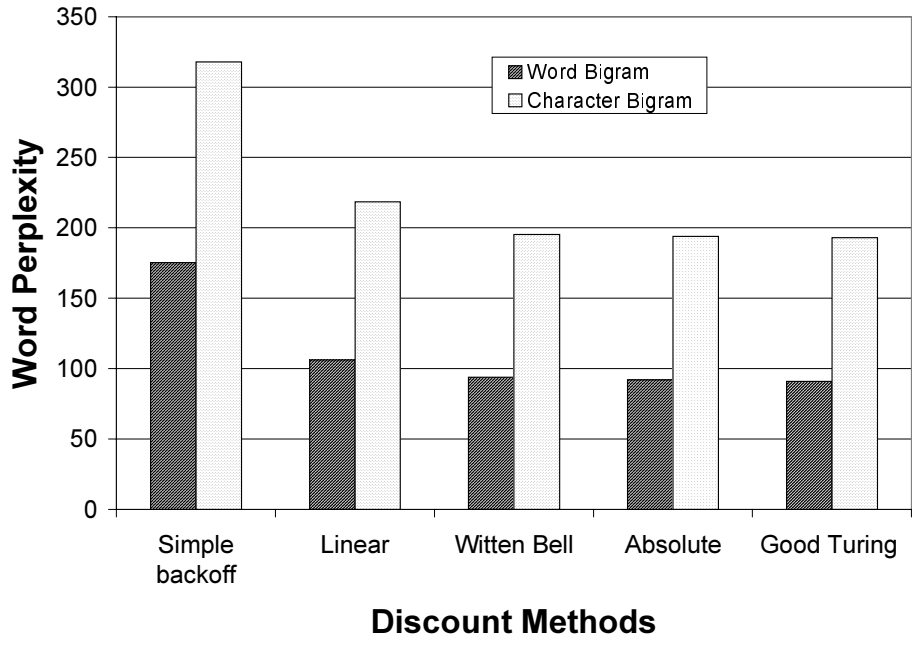

Figure 18: Comparison of different Discounting Methods for the Call Home task

5.6 Continuous Word Speech Recognizer

To examine the contribution of the proposed smoothing methods to the recognition accuracy, we have incorporated our language models into our continuous speech recognizer. Having reviewed some typical art-of-the-state speech recognition system, the chosen system is designed using Hidden Markov Models. However, there are only a few systems with similar parameters for comparison, such as IBM Call Home experiments.

5.6.1 Experiment Setup

All recognizers in the experiments used 408 base syllable as speech units. The architectures of the recognizers in all experiment are the same: they all use the hidden Markov Model (HMM) technique for acoustic modeling. All 408 HMMs has 8-states and 8 mixtures. The HMM states are arranged in a left-to-right, no-state-skipping topology. The segmental k-means algorithm is used for training and the Viterbi algorithm is used for decoding. 13 MFCC, 13 ΔMFCC, Energy and an ΔEnergy are used as feature vectors to the recognizer. Since our main theme is to compare the language models, no context-depend models are made to improve the recognition result.

Details of the different Language Models

1. No Language Model (No-LM): All of 408 base syllable are output candidate, with grammar network [sil] <Base_syllabe> [sil].
2. Fair Character Bigram (FCB-LM), Fair Word Bigram (FWB-LM): Language model is formed base on the Call Home Training Transcription (170K words)
3. GT Smoothed Fair Word Bigram (SFWB-LM): FWB-LM with smoothing techniques. Please refer to section 5.3.3 for the details the implementation.

4. Interpolated & GT Smoothed Fair Word Bigram (ISFWB-LM): Language model is formed by interpolating fair Call Home language Model and Hub5 language model (250K Words). Please refer to section 4.8 for details of the implementation.

5. Cheating Word Bigram (CWB-LM): Language model form base on Call Home testing transcription (48K Words).

5.6.2 Experiment Results:

All the five experiment results are shown in Table 45 and Figure 19. The percent accuracy is calculated by taking into account all of the deleted, inserted, and substituted words as given in the following equation:

$$error = \frac{sub.err + del.err + ins.err}{total \quad words}$$

$$\%Accuracy = (1 - \%error) * 100$$

Thus, it is possible to have an error rate greater than 100% or negative accuracy. While these error rates are much higher than for tasks involving read speech, they are comparable to the initial results obtained from the IBM research center.

By using the simple word bigram language model, we have the syllable accuracy increased from 20.17% to 27.24%, which is 35% improvement compared with system without language model. In addition, the Good Turing smoothed word bigram produces the syllable accuracy 31.46%. It is a significant improvement when compared with the simple backoff character bigram. Moreover, the interpolated language model can further push the accuracy up to 32.02%, which is 0.8% improvement over the smoothed one. Furthermore, we notice that if the perplexity is lowered to 18.35 (by cheating method), we can have 38.28% syllable accuracy.

These results are not as good as current state-of-the-art speech recognizer system for a couple of reasons. First, instead of using the context-dependent initial-final model, we use the isolated base syllable acoustic models, which may not be well trained in data insufficient situation. Second, we do not embed any tonal information in our speech recognizer, which is assumed able to settle some recognition ambiguity.

Language Model	Word Perplexity	Syllable Accuracy (%)	Character Accuracy (%)
No LM	N/A	20.17%	N/A
FCB-LM	N/A	27.24%	23.25%
FWB	175.13	29.37%	25.24%
SFWB-LM	90.88	31.75%	27.94%
ISFWB-LM	88.90	32.02%	28.13%
CWB-LM	18.35	38.28%	36.09%

Table 45: Recognition results for different language models for the Call Home task

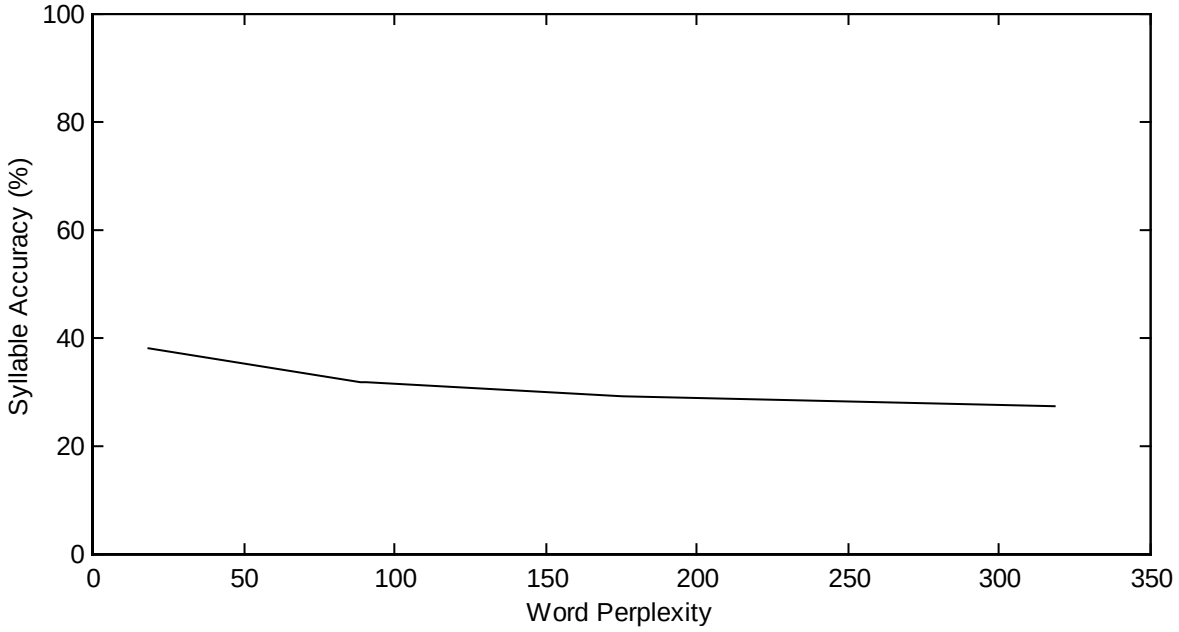

Figure 19: Recognition results for different language models for the Call Home task

5.7 Chapter Conclusion

In this chapter, we study the effect of various smoothing techniques in n-gram language models of Chinese. Detailed algorithm and its underlying inspiration are also presented. The Good Turing smoothing method give the best perplexity result, which is 90.88 for word language model. Moreover, the Good Turing smoothed bigram language model gives 31.46% syllable accuracy for the Call Home recognition task. On the other hand, the absolute and linear smoothing methods are easier to implement. It may then useful for simple application. Moreover, we can achieve 38.28% syllabi accuracy, if the perplexity is lowered to 18.35.

All the language models in this thesis are trained by full set of the Call Home Mandarin training transcription. However, there are only 127 thousand words in the training data and we have 33 million possible bigrams. We expect that the result can be further improved if a larger training corpus is available.

Chapter 6 Summary and Conclusions

In this final chapter, I summarize the results from the previous chapters. Next, I will describe some possibilities for improvement for the system and its components. Finally, I will suggest how the work described in the thesis might be forward.

6.1 Summary

The aim of this thesis is to study lexical access and language modeling in Mandarin speech recognition. Lexical access attempts to provide a speech recognizer a small subset of potential word from a large vocabulary, so that the searching speed for the recognizer can be dramatically increased. While statistic language model tries to capture and characterize the syntax constraints in a language. Proper language modeling is crucial to the performance of a speech recognition system.

Fundamental theory for the works in the thesis is described in chapter 2. The characteristic for Chinese is highlighted, which may be useful for western readers. Also the background theories for the components of our experimental recognizer is also described, such as acoustic modeling, recognizer search algorithm, statistical language model and smoothing techniques.

Chapter 3 presents our analysis on the lexical access by broad class features. The Mandarin broad class analysis is compared with English and Cantonese. It is found that the Mandarin broad class representation can uniquely specific 19% words in a 44404-word lexicon, and the expected cohort size is only 62.4. Thus, a subsequence recognizer only need to search 64.2 words instead of 44404 words. It is also marked that the percentage of uniquely specified word (UNIQ) of 6 broad classes are very similar for the languages, they are 19.0%, 16.7% and 15.7% for Mandarin, Cantonese and English respectively.

In chapter 4, two different kinds of language modeling approaches are studied. They are character level language model and word level language model. It was found that at the same level of OOV,

the word level language model gives a lower perplexity than character language model. This suggests that word level language model should be more appropriate for speech recognition tasks.

Smoothing methods for improving the language models are introduced in chapter 5. Four different kinds of smoothing techniques are compared. They are Witten-Bell, Good-Turing, absolute and linear smoothing. Detailed algorithm and its underlying inspiration are also presented. It is found that the Good-Turing algorithm give the lowest perplexity results for our Call Home task. An experimental continuous Mandarin speech recognizer is also constructed to test the real recognition performance for the language models. The system achieves 38.28% syllable accuracy when the word perplexity is at 18.35.

6.2 Further Work

In this section, we will mention some possible future extensions of this work.

While the lexical access method is proved an effective way for a fast search recognizer. It may not be practical in the real recognition task. For example, the recognition result for the broad class model may not be good, which causes the appropriate candidate is not passed to the next level detailed recognizer. Finally, this leads to the recognizer error. Therefore, a large isolated word recognizer must be built to test the performance of lexical access.

The language models used in our experiments are in character and word level only. It may be interesting in building a syllable or morpheme level language model.

The smoothing methods used in this thesis tries to estimate the probability of unseen n-gram by the counts of seen n-gram. However, it losses some linguistic information. We can have a better estimation by checking the n-gram starting with its synonyms. For example, the probability of an unseen word pair w_1w_2 can be expressed as: $p(w_1w_2)=p(w_sw_2)$ where w_s = arg max Sim(w_1,w) and Sim(w_1,w) is a similarity function to express the similarity between the two words. The similarity of the two words can be represented as a square matrix A. The elements of this matrix are the probability of in which a word followed by other words, i.e. the element a_{ij} is the probability of word w_i follow by word w_j. We can than compare different columns in the matrix to find the similarity.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{21} & \dots & a_{2n} \\ . & . & . & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

6.3 Conclusion

The major work presented in this thesis has contributed to two aspects. The first one is the study of lexical access for large isolated word speech recognition. The construction of lexical access models that attempt to speed up a large vocabulary isolated word recognizer. The second one is the study of language model for large vocabulary spontaneous speech recognition. The language model and its smoothing method provide a powerful tool for predicting next word for a recognizer. The results of this thesis show that linguistic knowledge is certainly beneficial to speech recognition.

Reference

[1] Fromkin & Rodman, "An Introduction to Language", Harcourt Brace Jovanovich College Publishers, 1993.

[2] Stephen Matthews and Virginia Yip, "Cantonese: A Comprehensive Grammar", Routledge, 1994.

[3] Suzanne Berger & Richard K.Lester, "Made by Hong Kong", Oxford, 1997.

[4] L.S. Lee, "Voice Dictation of Mandarin Chinese", IEEE Signal Processing Magazine, p63-101, July, 1997.

[5] Roe, D.B.; Wilpon, J.G., "Whither speech recognition: the next 25 years", IEEE Communications Magazine, Nov, 1993.

[6] Benjamin Tsou, "Some Characteristics of Chinese Language", Language Information Sciences Research Center Newsletter 2, City University of Hong Kong, 1997.

[7] Reddy D.R. et al. "Knowledge and its Representation in a Speech Understanding System", Knowledge and Cognition: Ninth annual Symposium of Cognition, pp.253-285, 1974.

[8] Lee K.F. “Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX system”, Ph.D. Thesis, Computer Science Department, Carneigie Mellon University, Apr. 1988.

[9] Hwang Mei-Yuh. “Sub-phonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition”, Ph.D. Thesis, Technical Report No. CMU-CS-93-230, Computer Science Department, Carneigie Mellon University, Dec 1993.

[10] John R. Deller, JR, John G. Proakis, John H. L. Hansen, "Discrete-Time Processing of Speech Signals", Prentice Hall, 1993.

[11] L.R. Rabiner, R.W. Schafer, "Digital Processing of Speech Signals", Prentice Hall, Englewood Cliffs, N.J., 1978.

[12] X.D. Huang, Y. Ariki. M.A. Jack, "Hidden Markov Models for Speech Recognition", Edinburgh University Press, 1990.

[13] Steve Young, "A review of large-vocabulary continuous-speech recognition", IEEE Signal Processing Magazine, September 1996.

[14] Richard J. Mannone, Xiaoyu Zhang, Ravi P. Ramachandran, "Robust Speaker Recognition", IEEE Signal Processing Magazine, September 1996.

[15] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", IEEE Trans , Acoustic, Speech and Signal Proc., vol 29, No2, pp.254-272, 1981.

[16] M.J.F. Gales, S.J. Young, "Cepstral Parameter Compensation for HMM Recognition in Noise", Speech Communication, Vol 12, No 3, pp. 231-239, 1993.

[17] L.R. Rabomer, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol 77,pp. 257-286, Feb, 1989.

[18] S. Austin, R. Schwartz, "The Forward-Backward Search Algorithm for continuous Speech Recognition", Proc ICASSP, S10.3, Toronto, 1991.

[19] E. Bocchieri, "A Study of Bean search Algorithm for Large Vocabulary Continuous Speech Recognition and Methods for Improved Efficiency", Proc Eurospeech, pp.1521-1524, Berlin, 1993.

[20] G.J. Lidstone, "Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities", Transaction of the Faculty of Actuaries, 8:182-192, 1920.

[21] P.C. Woodland, C.J. Leggetter, J. Odell, V.Valtchev, S.J. Young, "The 1994 HTK Large Vocabulary Speech Recognition System." ", International Conference of Acoustics, Speech, and Signal Processing, Vol 1,pp 73-76, Detroit, 1995.

[22] David W. Shipman and Victor W. Zue, "Properties of Large Lexicons: Implications for advance isolated word recognition systems", International Conference on Acoustics, Speech and Signal Processing, 1982.

[23] D.P. Huttenlocher & V.W. Zue, "A model of lexical access from partial phonetic information", International Conference on Acoustics, Speech and Signal Processing,, 1984.

[24] D.P. Huttenlocher & V.W. Zue, "Phonotactic and Lexical Constraints in Speech Recognition", Proc. American Association for Artificial Intelligence Conference, pp.172-176, 1983.

[25] Luciano Fissore, Pietro Laface, Giorgio Micca & Roberto Pieraccini, "Lexical Access to Large Vocabularies for Speech Recognition", IEE Transaction on Acoustics, Speech and Signal Processing, VoL.37, No.9, 1989.

[26] Jialu Zhang, "On the Syllable Structures of Chinese relating to Speech Recognition", International Conference of Spoken Language Processing, 1996.

[27] Hasan, H.; Pardo, J.M.; Alexandres, S.; Casado, C. , "Phonetic properties of a large Spanish lexicon and its implications for large vocabulary speech recognition", International Conference of Acoustics, Speech, and Signal Processing, 1989.

[28] O'Grady, W., M. Dobrovolsky & M. Aronoff, "Contemporary Linguistics. An Introduction", Longman, 1997.

[29] Jannedy, Stefanie, Robert Poletto, and Tracey Weldon, "Language Files: Materials for an Introduction to Language", Ohio State University Press, 1991

[30] Make use of syllable structure and phonotactic rules in Mandarin. Jialu Zhang (1996). On the syllable structures of Chinese relating to speech recognition. International Conference of Spoken Language Processing, ICSLP, 1996.

[31] L.R.Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer. "A Tree-Based Statistical Language Model for Natural Lanugage Speech Recognition." IEEE Trans ASSP, Vol 37, No.7,1989.

[32] E.Black, F.Jelinek, J.Lafferty,D.M.Magerman, R.Mercer,S.Roukos, "Towards History-based Grammars: Using Ricker Models for Probabilistic Parsing", Proc DARPA, Spoken Language systems Workshop, Feb 1992.

[33] S. Deligne, F. Bimbot, "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams." ", International Conference of Acoustics, Speech, and Signal Processing, Vol 1, pp.169-172, Detroit, 1995.

[34] Lawrence Rabiner & B.H. Juang, "Fundamentals of Speech Recognition", Prentice Hall, 1993.

[35] E.L. Lehmann, "Theory of Point Estimation", New York, John Wiley & Sons, 1983.

[36] L.R. Bahl, F. Jelinek, & R.L. Mercer, "A Maximum Likelihood approach to continuous Speech Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(2): 179-190, 1983.

[37] T. Crystal, M. Cowing, A. Martin, and D. Pallett, "LVCSR", Proceedings of Speech Research Symposium, P.P 21-39, June 1999.

[38] R. Agarwal, B.Wheatley, Y. Muthusamy, and T.Staples, "Diagnostic Profiling for Speech Technology Development: Call Hone Analysis", Proceedings of Speech Research Symposium, P.P 131-137, June, 1995.

[39] Fu-Hua Liu, Micheal Picheny, Patibandla Srinivasa, Michael Monkowski and Julian Chen, "Speech Recognition on Mandarin Call Home: A large-Voculary, Conserational, and Telephone Speech Corpus", International Conference on Spoken Language Processing, 1996.

[40] J.Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research Development", ICASSP-92, pages I-517-520, March 1992.

[41] L.S. Lee, et al, "Golden Mandarin(I)-a real time Mandarin Speech Dictation Machine for Chinese Language with Very Large Vocabulary", IEEE Trans. on Speech and Audio Processing, vol. 1, no. 2 P.P. 158-179, April 1993.

[42] Kyuwoong Hwang, "Vocabulary Optimization Based on Perplexity", IEEE Trans ASSP, 1997.

[43] R. Rosenfeld, "Adaptive Statistical Language Modeling: A Statistical approach", PhD Thesis, School of Computer Science, Carnegie Mellon university, April 1994.

[44] A. Nadas, "On Turing's formula for word probabilities", IEEE Trans Acoustics, Speech and Signal Proc., vol 33, pp 1414-1416, Dec 1985.

[45] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech and Signal Processing, ASSP-35:400-401, 1987.

[46] I. J. Good and G.H. Toulmin, " The number of new species, and the increase in population coverage, when a sample is increased," Biometrika, vol. 43, Pts. 1 and 2, pp. 45-63, June. 1956.

[47] Ian H. Witten & Timothy C. Bell, "The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression", IEEE Transactions on Information Theory, Vol 37, No.4 July 1991.

[48] H. Ney, U. Essen & R. Kneser, "On structuring probabilistic dependencies in stochastic language modeling", Computer Speech and Language, vol 8, p1-38,, 1994.

[49] S.M. Ross, "Introduction to Probability Models", Academic Press, London, 1985.

[50] W. Gale and K. Church, "What is wrong with adding one?", Corpus-based Research into Language, Amsterdam, 1994.

[51] I. B. Efron and R. Thisted, "Estimating the number of unseen species: How many words did Shakespeare know?" Biometrika, vol. 63, no. 3, pp. 435-447, 1976.

[52] Timothy C. Bell, J.G. Cleary & Ian H. Witten, "Text Compression", Englewood Cliffs, N.J, Prentice-Hall, 1990.

[53] Good I.J., "The population frequencies of species and the estimation of population parameters.", Biometrika,40:237-264, 1953.

[54] Frederick Jelinek, "Statistical Methods for Speech Recognition", MIT Press, 1997.

[55] Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Prentice Hall, 1999.

J.K. Skwirznski, ed. Dordrecht, The Netherlands: Nijhoof, 1985.

[56] Hermann Hey, Ute Essen & Reinhard Kneser, "On the Estimation of 'Small' Probabilities by Leaving-One-Out", IEEE Transactions of Pattern Analysis & Machine Intelligence, vol. 17, No.12, p1202-1212, Dec. 1995.

[57] A. Nadas, "Estimation of probabilities in the language model of the IBM speech recognition system", IEEE Trans, Acoustic, Speech and Signal Proc., vol.32, pp.859-861, Aug. 1984.

[58] Jelinek, F. "Direct Parsing of Text, Image Models and Their Speeech Model Cousins", Springer-Verlag, 1996.