Unsupervised Clustering and Automatic Language Model Generation for ASR

by

Sushil Kumar Podder

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The goal of an automatic speech recognition system is to enable the computer in understanding human speech and act accordingly. In order to realize this goal, language modeling plays an important role. It works as a knowledge source through mimicking human comprehension mechanism in understanding the language. Among many other approaches, statistical language modeling technique is widely used in automatic speech recognition systems. However, the generation of reliable and robust statistical model is very difficult task, especially for a large vocabulary system. For a large vocabulary system, the performance of such a language model degrades as the vocabulary size increases. Hence, the performance of the speech recognition system also degrades due to the increased complexity and mutual confusion among the candidate words in the language model.

In order to solve these problems, reduction of language model size as well as minimization of mutual confusion between words are required. In our work, we have employed clustering techniques, using self-organizing map, to build topical language models. Moreover, in order to capture the inherent semantics of sentences, a lexical dictionary, WordNet has been used in the clustering process.

This thesis work focuses on various aspects of clustering, language model generation, extraction of task dependent acoustic parameters, and their implementations under the framework of the CMU Sphinx3 speech engine decoder. The preliminary results, presented in this thesis show the effectiveness of the topical language models.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Fakhri Karray, for guiding me through this research. His enlightening discussions, valuable suggestions, and strict guidance helped me to complete my research work. Throughout my term as a graduate student, he has been a source of knowledge and trusted advice.

I have a special debt to my co-supervisor, Professor Otman Basir for his commitment to my research work and for his continuous advice. Thanks are also due to professor Mohamed Kamel, director of PAMI lab for his support.

To perform research and doing experiments, all of my colleagues, especially Dr. Jiping Sun, director of Vestech Inc. helped me a lot. I would like to thank him for his suggestions and supports.

Finally, I am grateful to all of my family members. In special, I like to thank my wife, Mallika for her wholehearted support during the undertaking of this research.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Natural language is a complicated phenomenon that developed over a long period of time through using organs and brain structures of human beings. We, human beings use grammatical rules subconsciously when we process and try to realize meaning of a spoken language. But machine cannot understand all of the nuances of human communication. In order to understand human language, it needs explicit knowledge about the semantic as well as syntactic structure of the message. This is the language model, whose function is to supply such kind of knowledge to deal with the variability and uncertainly in natural language.

In order to process natural language, two major approaches have been developed: one is rule-based approach, and the other is probabilistic approach. The rule-based approach is developed based on the syntactic and semantic structure of the sentence. In this approach, experts hand craft a set of rules through analyzing a huge amount of sample sentences, and these rules are used for analyzing the texts whose structure is not known. But due to the variability and uncertainty in natural language, rule-based approach is used only for very specific applications (usually domain specific) with a limited vocabulary.

The probabilistic approach is more flexible than rule-based approach and it is capable of capturing the statistical regularities of natural language. In this approach, the word's contextual information is embedded during the modeling process. It is usually used for a large vocabulary domain independent system.

Language model plays an important role in various applications as diverse as handwriting recognition, spelling and grammar correction, machine translation, part-of-speech tagging, and others. But its main application is in speech recognition that is presented in subsequent chapters.

1.1 Basis of the Speech Recognition Systems

Speech recognition is a technology that enables the computer to convert human speech into text. In order to achieve this goal, several approaches, namely template-based approach [1], knowledge-based approach [**Error! Reference source not found.**,3,4], probabilistic approach [4,5,6,7,8,9,10,11], connectionist approach [12], etc., have been developed and tested over a long period of time.

The state of the art speech recognition system is a combination of various technologies including signal processing, signal modeling, pattern matching, and many other techniques used in AI. A typical ASR system consists of three main components, namely, signal processing, acoustic modeling, and language modeling.

Speech recognition is a difficult task. In order to actualize the goal of the speech recognition, a lot of effort is given for building a good decoder, which comprises of two main modules: acoustic modeling module, and language modeling module. This effort is termed as *training*. Depending on the task, basic unit of speech is decided in this phase. Through collecting huge speech data, acoustic modeling is done using several techniques like DTW [4,13], hidden Markov model (HMM) [4,5,10], Neural network [14], support vector machine [15], and etc. Beside these, grammar and lexicon generation are performed in this stage. Once model generation is completed, next stage is pattern matching, which is known as *testing* or recognition phase. In this phase, an unknown input speech utterance is matched with the stored models or templates and identified through comparing the overall scores of templates or models.

In the following subsection, basic components of ASR system are discussed from the viewpoint of HMM.

1.1.1 Signal processing

Signal processing is one of the important steps of the ASR system. The main goal of the signal processing is to extract some representative features in speech. Hence, the signal processing task sometimes called as *feature extraction*, or *front-end processing*.

Today's state of the art speech recognition systems are using a wide variation of front-end signal processing techniques. However, basic front-end processor comprises of two main components as shown in Fig.1.1.

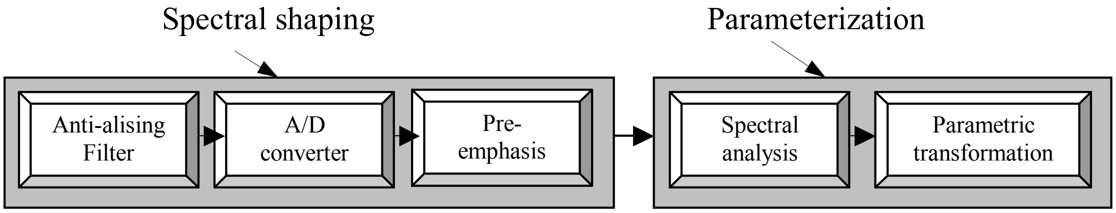


Figure 1-1: The main components of signal processing

- Spectral shaping [16] performs three basic operations:
  - Anti-alising filtering: An analog filter that filters out all of the unwanted frequency components before digitization is called anti-alising or pre-sampling filter. Actually anti-alising filter ensures the signal quality through maintaining Nyquist's sampling rate criteria.
  - A/D converter: A process in which analog signal is converted to digital signal.
  - Pre-emphasis filter: A finite impulse response (FIR) filter that emphasis important frequency components in a speech signal.
- Parameterization [16] is the way to represent the speech signal by the use of parameter, which is more convenient for subsequent processing.

- Spectral analysis: A process of estimating the frequency response of vocal tract through analyzing time domain or frequency domain characteristics of the speech signal. Spectral analysis is one of the crucial parts of speech recognition systems, which partially affects the performance of ASR system. Some of the popular spectral analysis algorithms are listed in Tab. 1.1:
- Parametric transformation [16]: In order to capture the spectral dynamics, parameter transformation plays an important role. It provides temporal information in HMM through computing delta or delta-delta coefficients from the aforesaid parameters.

Table 1-1: Spectral analysis algorithms

Speech production motivated representation	Perceptually motivated representation
Digital Filer Bank	Mel-frequency cepstral coefficients (MFCC)
Fourier Transform Filter Bank	Bilinearly transformed cepstrum
Linear predictive coding (LPC)	Perceptual linear prediction (PLP)
LPC derived filter bank amplitudes	
LPC derived cepstral coefficients	
Fourier transform ceptral coefficients	

1.1.2 Acoustic modeling

Acoustic modeling [9,11] is the central part of the speech recognition system. It provides a way to encode speech features during the training process and detect and classify possible acoustic patterns during the recognition phase.

In order to conceive the basis of acoustic modeling, we need to introduce some simple mathematical formulation.

Let $\Theta = \theta_1, \theta_2, .., \theta_m$, an observation sequence derived from the speech signal. The goal of the speech recognizer is to find out most likely word sequence $W^* = w_1, w_2, .., w_n$, expressed as

$$\begin{aligned} W^* &= \arg\max_W P(W \mid \Theta) = \arg\max_W \frac{P(\Theta \mid W) \cdot P(W)}{P(\Theta)} \\ &\cong \arg\max_W P(\Theta \mid W) \cdot P(W) \end{aligned} \tag{1.1}$$

In order to estimate $P(\Theta \mid W)$, we need an acoustic model. Among many other techniques, hidden Markov model (HMM) is widely used because of its capability of modeling the temporal structure and variability in speech. This is a probabilistic pattern matching approach, capable of modeling a time sequence of speech as an output of stochastic process.

1.1.3 Language modeling

Acoustic modeling plays a vital role for encoding speech. For recognition and understanding the natural language, it plays a complementary role with language model. In order to understand the natural language we need lexical as well as syntactic knowledge. Language model provides such kind of information. In (1.1), $P(W)$ is a priori probability supplied by language model. The priori probabilities in language model are as important as acoustic model. These probabilities act as a contributing factor in final decision making from a set of hypothesis.

1.2 Motivation and Goals of the Research

Although statistical language models have some disadvantages, these are widely used in the speech recognition system. For a large vocabulary system, statistical language model performs well if it is built carefully with sufficient training data. But usually as the vocabulary size increases, search path and mutual confusion among words increase proportionally and consequently the estimation of the most likely words based on the previous history becomes challenging and sometimes unmanageable.

In order to solve these problems, reduction of language model size as well as minimization of mutual confusion between words are required. Brown et al. [17] has taken the approach of clustering believing that can play an important role in improving the language model. Recently, the speech community has begun to address the use of clustering in building better language model, and thus improve recognition accuracy. Florian and Yarowsky [18] utilized hierarchical and dynamic topic-based clustering to build language models.

The aforesaid works have motivated us to use clustering technique, especially self-organizing map, in building language model. Moreover, in order to capture inherent semantics of sentences, a lexical dictionary, WordNet [19,20] has been used in the clustering process.

Main goal of this thesis work is to enhance the recognition performance of the HMM-based speech recognizer so that it can be used in free dictation system.

This thesis work focuses on various issues of clustering process, language model generation, extraction of task dependent acoustic model parameters, and their implementations under the framework of CMU Sphinx3 [21].

1.3 Thesis Organization

In chapter 2 overview of the statistical language modeling technique is given, Chapter 3 presents the various aspect of data clustering and language model generation. The detail about the extraction of task dependent acoustic model parameters is presented in Chapter 4.

Preliminary experimental results are presented and discussed in chapter 5. Chapter 6 summarizes the main contributions, highlights the key issues covered, and recommends further developments.

Chapter 2

Overview of Statistical Language Modeling Techniques in ASR

In the speech recognition system, language model plays an important role in decision-making. It works as a knowledge source through mimicking human comprehension mechanism in understanding the language. Actually, language model provides a priori knowledge about the organization of text in a corpus at word level. Therefore, It is possible to determine the most likely word sequence out of multiple hypotheses and enhance the recognition performance.

Depending on the nature of application, different types of language models, namely, finite-state language model [11], grammar language model [11], and stochastic language model [9,10,11] are used in speech recognition system. Among them, stochastic modeling enjoys wide spread usage due to its elegant characteristic in modeling the language for large vocabulary speech recognition applications.

This chapter provides an introduction to the key terms and major concepts of stochastic language modeling. It starts by describing main source of statistics, measuring the quality of language model; next it describes various issues on generating n-gram language models from the sparse data.

2.1 Stochastic Language Models

Stochastic language model is based on the concept of Markov process, where a word sequences assumed to be generated from an n-order Markov source. In the ASR system, stochastic modeling provides wonderful solution to the problem of huge search space. Through providing adequate contextual information in terms of probability, it makes the ASR system more efficient and accurate.

n-gram language model is an example of stochastic language model, which is widely used in the HMM based ASR system.

2.1.1 n-gram language model

In the n-gram language model, probability of a word string, $W = w_1, w_2, ..., w_n$, i.e., $P(W)$ is estimated using Bayesian formula as follows:

$$\begin{aligned} P(W) &= P(w_1, w_2, ..., w_n) \\ &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)...P(w_n \mid w_1, w_2, ..., w_{n-1}) \qquad (2.1) \\ &= \prod_{i=1}^{n} P(w_i \mid w_1, w_2 ...., w_{i-1}) \end{aligned}$$

According to (2.1), v^i different values (where v is the vocabulary size) are to be estimated to compute $P(W)$. But in reality, estimation of such a huge set of probabilities is impossible, even for moderate values of *i*. In order to overcome this drawback, probability of $P(w_i)$ is estimated considering only preceding $N-1$ events. This gives birth a new concept, called *n*-gram language model.

For practical purpose, the value of *N* is usually limited up to two. When the model considers no previous event, model is called uni-gram model. Same way, when it considers previous one and two events, model is called bi-gram and tri-gram respectively.

For uni-gram, probability of each word is computed based on the relative frequency estimation:

$$P(w_i) = \frac{C(w_i)}{\sum_{\forall i} C(w_i)} \tag{2.2}$$

Where $C(w_i)$ is the total count of w_i in the training data set.

In the bi-gram model, probability of appearing a word just depends on the preceding word. Mathematically, it can be expressed and computed as:

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{\sum_{w_i} C(w_{i-1}, w_i)} \tag{2.3}$$

In the tri-gram model, probability of a word depends on two preceding words. It is computed using the same relative frequency estimation as follows:

$$P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{\sum_{w_i} C(w_{i-2}, w_{i-1}, w_i)} \tag{2.4}$$

Among different choices, tri-gram language model is widely used in most of the HMM based ASR systems. In the subsequent section, problem of tri-gram estimation from the sparse data are discussed.

2.2 Measuring Model Quality

Since the performance of ASR system depends partially on the performance of language model, generating an efficient and robust language model is one of the critical issues in the speech recognition system. In practice, efficiency of the language mode is evaluated based on the error rate of the recognizer. Usually, a good language model provides good recognition performance. But, such kind of evaluation is expansive and time consuming. Alternatively, there is an in-expansive way to measure the quality of language model based on the concept of information entropy, devised by Shannon in the early days of information theory. Since, language model can be treated as an information source, Shannon theory can easily be expanded to measure the information entropy of language model.

In the following subsection, one of the widely used measures termed as Perplexity [9, 10, 11] (derived from the concept of entropy) is presented.

2.2.1 Perplexity

As stated earlier that quality of the language model is evaluated through measuring the information entropy (entropy is the measure of average uncertainly of a language event). Conventionally, this evaluation measure is expressed by the term “perplexity”, which is defined as:

$$perplexity = 2^{entropy}$$

The entropy of a *n*-gram language model, with following specifications, is estimated as:

$W = \{w_1, w_2, ..., w_n\}$: A word sequence, of length *n*

$P(W)$: Probability of a word sequence W

$H(W)$: Entropy (wise to say cross-entropy)

$$H(W) = -\frac{1}{n}\log_2 P(W) \tag{2.5}$$

So perplexity, $PP(W)$ is defined as

$$PP(W) = 2^{H(W)} \tag{2.6}$$

Since perplexity is highly correlates with recognition performance, low perplexity (model in which number of words branching from a preceding word is lower in average) model is desirable.

2.3 Language Model Smoothing Techniques

n-gram language model performs well when it is trained with sufficient training data. But, in practice, training data is always insufficient (IBM researchers investigate a fact that about 23% of tri-gram events are unseen in the training data set [10]. Other research on an ATIS database shows that only 1.28% bi-grams, appearing in a testing dataset are available in the training dataset [9], and 45% of bi-grams appears only once in the training dataset). Of course, it is not practically possible to cover all words that might appear in the real world. This phenomenon, addressed as data sparseness, is one of the key problems in estimating *n*-gram probabilities reliably. Smoothing is a technique that is successfully able to encounter this problem. Actually, Smoothing provides a way to generalize the language model through adjusting low probabilities of the rear events upward and high probabilities downward.

In order to explain the smoothing technique, Lets we consider a simple bi-gram event $(w_i \mid w_{i-1})$, whose probability is estimated as:

$$P(w_i \mid w_{i-1}) = \frac{C(w_i, w_{i-1})}{\sum_{w_i} C(w_i, w_{i-1})} \tag{2.7}$$

According to (2.7), $P(w_i \mid w_{i-1}) = 0$, for unseen data. In order to avoid the zero probability situations, we can consider following simplest solutions:

Solution 1: Add threshold smoothing

$$P(w_i \mid w_{i-1}) = \begin{cases} \dfrac{C(w_i \mid w_{i-1})}{\sum_{w_i} C(w_i \mid w_{i-1})} & \text{if } \; C(w_i|w_{i-1}) > 0 \\ \theta & \text{otherwise} \end{cases} \tag{2.8}$$

Solution 2: Add one smoothing

$$P(w_i \mid w_{i-1}) = \frac{1 + C(w_i, w_{i-1})}{\vartheta + \sum_{w_i} C(w_i, w_{i-1})} \tag{2.9}$$

Where ϑ, size of the vocabulary

Solution 3: Additive smoothing

$$P(w_i \mid w_{i-1}) = \frac{\zeta + C(w_i, w_{i-1})}{\zeta\vartheta + \sum_{w_i} C(w_i, w_{i-1})} \quad \text{where} \begin{cases} \zeta = 1 \text{ for Laplace law} \\ \zeta = 0.5 \text{ for Lidstone's law} \end{cases} \tag{2.10}$$

The solutions stated above work unreliably in practice. Instead, there are some mathematically sophisticated practical approaches available in the wide range of literature. In the following subsection, some of the popular approaches are presented briefly.

2.3.1 Good-Turing estimator

Good-Turing probability estimation [9,10,22] is a smoothing technique that deals with rear *n*-grams. Actually, this estimator performs well for an infinite corpus and infinity language [9]. For *n*-gram smoothing, parameter of the language model is smoothed based on their frequency. Although, a wide variety of Good-Turing estimators are used to deal with the finite corpus, all of the variants use the following equation to estimate the probability of an event κ that appears φ times is:

$$P(\kappa) = \frac{\varphi + 1}{T} \cdot \frac{\eta_{\varphi+1}}{\eta_{\varphi}} \tag{2.11}$$

Where, η_φ is the number of the *n*-gram that appear exactly φ times in the training data, and T is the total number of observations, given by:

$$T = \sum_{\varphi=0}^{\infty} \varphi \cdot \eta_\varphi \tag{2.12}$$

The effectiveness of *n*-gram smoothing using Good-Turing estimator will be more visible, if we compare (2.11) with following equation that estimates the probability of same event by relative frequency estimation as:

$$P(\kappa) = \frac{\varphi}{T} \tag{2.13}$$

With (2.13), $P(\kappa)$ = 0 for φ = 0, but with (2.11), $P(\kappa) >$ 0 for φ = 0. Thus Good-Turing estimator provides smoothing operation through ensuring the probability of the unseen events as nonzero.

2.3.2 Kaltz smoothing

Although Good-Turing estimate is the key tool of several smoothing technique, it by itself suffers from performance degradation due to its inherent weakness of not being able to combine the higher order models with lower order models. Katz smoothing technique overcomes this weakness.

Katz technique [11,23] is simply an extension of Good-Turing technique, which combined the interpolation between higher order and lower order *n*-grams. Katz smoothing technique for bi-gram is summarized as follows:

$$P_{Katz}(w_i \mid w_{i-1}) = \begin{cases} C(w_{i-1}w_i)/C(w_{i-1}) & \text{if } \varphi > k \\ q_\varphi C(w_{i-1}w_i)/C(w_{i-1}) & \text{if } \mathrm{k} \geq \varphi > 0 \\ \beta(w_{i-1})P(w_i) & \text{if } \varphi = 0 \end{cases} \tag{2.14}$$

In (2.14), q_φ is the discount ratio, which is expressed as:

$$q_\varphi = \frac{\dfrac{\varphi^*}{\varphi} - \dfrac{(k+1)\eta_{k+1}}{\eta_1}}{1 - \dfrac{(k+1)\eta_{k+1}}{\eta_1}} \tag{2.15}$$

$\beta(x)$ is the scaling factor, which is expressed as:

$$\beta(w_{i-1}) = \frac{1 - \sum_{w_i:\varphi>0} P_{Katz}(w_i \mid w_{i-1})}{1 - \sum_{w_i:\varphi>0} P(w_i)} \tag{2.16}$$

φ is the frequency of bi-gram and φ^* is the discounted frequency, which is expressed as:

$$\varphi^* = (\varphi+1)\frac{\eta_{\varphi+1}}{\eta_{\varphi}} \tag{2.17}$$

k is the threshold, whose value is recommended as 5~8.

2.3.3 Kneser-Ney smoothing

There are many other ways to discount the probability from the probability of nonzero events. Kneser-Ney smoothing [24] is one of these techniques that expands the notions of discounting with back-off models. Here below is the algorithm of Kneser-Ney for bi-gram smoothing:

$$P_{KN}(w_i \mid w_{i-1}) = \begin{cases} \dfrac{\max\{C(w_{i-1}w_i) - D, 0\}}{C(w_{i-1})} & \text{if } C(w_{i-1}w_i) > 0 \\ \beta(w_{i-1})P_{KN}(w_i) & \text{otherwise} \end{cases} \tag{2.18}$$

In (2.18), $P(w_i)$ is defined as

$$P(w_i) = \frac{\widehat{C}(w_i)}{\sum\limits_{w_i} \widehat{C}(w_i)} \tag{2.19}$$

Where $\widehat{C}(w_i)$ denotes the number of unique words that follow w_i, and $\beta(w_{i-1})$ is the scaling factor that makes the probability distribution sum to 1, and expressed as:

$$\beta(w_{i-1}) = \frac{1 - \sum\limits_{w_i : C(w_{i-1}w_i) > 0} \dfrac{\max\{C(w_{i-1}w_i) - D, 0\}}{C(w_{i-1})}}{1 - \sum\limits_{w_i : C(w_{i-1}w_i) > 0} P(w_i)} \tag{2.20}$$

2.3.4 Deleted interpolation smoothing

Although Backing-off smoothing techniques, described above are feasible to implement, interpolated techniques are more effective (especially than Good-Turing smoothing [9]) and widely used in state of the state of the art speech recognition system.

Deleted interpolation smoothing technique [9] is based on the concept of interpolated models that leads to compute a probability of an event from the weighted average of some distributions as:

$$P_I(\kappa) = \lambda_1 P_1(\kappa_1) + \lambda_2 P_2(\kappa_2) \tag{2.21}$$

Where κ, an event, whose probability to be computed, $P(\kappa)$ is the distribution, λ_i is weight, whose value must be nonzero and $\lambda_1 + \lambda_2 = 1$.

For bi-gram, smoothed probability is computed through interpolating a bi-gram and uni-gram probabilities linearly as follows:

$$P_I(w_i \mid w_{i-1}) = \lambda_1 P_1(w_i \mid w_{i-1}) + \lambda_2 P_2(w_i) \tag{2.22}$$

Where $P(w_i \mid w_{i-1})$ is computed using kept-data (a larger set of training data set) based on the relative frequency approach as:

$$P(w_i \mid w_{i-1}) = \frac{C(w_i, w_{i-1})}{C(w_i)} \tag{2.23}$$

In the above equation, weight λ_i is estimated from the held-out data (a smaller set of training data set)

In the deleted interpolation smoothing technique, uni-gram and bi-gram probabilities are both considered when the bi-gram event is unseen or infrequent.

For the tri-gram model, smoothing is done through interpolating tri-gram, bi-gram and uni-gram probabilities as:

$$P_I(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1 P_1(w_i \mid w_{i-2}, w_{i-1}) + \lambda_2 P_2(w_i \mid w_{i-1}) + \lambda_3 P_3(w_i) \tag{2.24}$$

Where, $P(w_i \mid w_{i-2}, w_{i-1})$, $P(w_i \mid w_{i-1})$, and $P(w_i)$ are estimated from the kept-data, and λ_i $(1 \le i \le 3)$ are estimated from the held-data set.

Chapter 3

Data Clustering for Generating Topical Language Models

Statistical language model is most widely used for large vocabulary ASR system. The job of the *n*-gram statistical language model is to impose some grammatical constraints so that correct word sequence appears in the ASR output. However, systems using such kind of model suffer acute performance degradation once the vocabulary size exceeds some critical value. For large vocabulary systems, search path and mutual confusion among the words increase proportionally and consequently estimation of the most likely words based on the previous history becomes sometimes unmanageable.

In order to reduce the complexity and mutual confusion in language model, downsizing (language model size) is therefore necessary for practical application. Data clustering using self-organizing map is deemed to be a promising solution.

Since WordNet, a lexical dictionary has a significant contribution in improving the clustering performance; it can be used in clustering process for improving the language model performance.

In this chapter, theoretical as well as practical aspects of data clustering, specifically, self-organized map is discussed first, next different steps including data preparation using WordNet and without WordNet, vector representation, self-organization and usage of clustered data for generating topical language models are discussed.

3.1 Data Clustering

Clustering algorithms partition a set of objects into groups or clusters [23,24,25]. There are two types of structures produced by clustering algorithms, hierarchical clustering and flat or non-hierarchical clustering. Flat clustering simply consists of a certain number of clusters and relation between clusters is often undetermined. Most algorithms that produce flat clustering are iterative. They start with a set of initial clusters and they are updated by iterating a reallocation operation that reassigns objects. A hierarchical clustering is a hierarchy with the usual interpretation that each node stands for a subclass of its mother's node. The leaves of the tree are the single objects of the clustered set. Each node represents the cluster that contains all the objects of its descendant.

Another important distinction between clustering algorithms is whether they perform a soft clustering or hard clustering. In a hard assignment, each object is assigned to one and only one cluster. Soft assignments allow degrees of membership and membership in multiple clusters.

3.2 Basis of Kohonen Self-Organized Map

The Kohonen self-organized map (SOM) [25,26] is one of most prominent and widely used artificial neural network algorithms. SOM uses unsupervised learning methodology that organize the data through discovering the hidden structure imbedded in it. The Kohonen SOM has a number of unique features [26] like, dimension reduction, high-dimensional data visualization, topographic mappings, etc.

Among these, topographical mapping and dimension reduction are very important features that make this algorithm useful for data clustering. Kohonen formulated the principle of topographic mapping based on the idea of the organization of cortex, a self-organized map in the human brain. Fig. 3.1 shows the feature-mapping model of Kohonen.

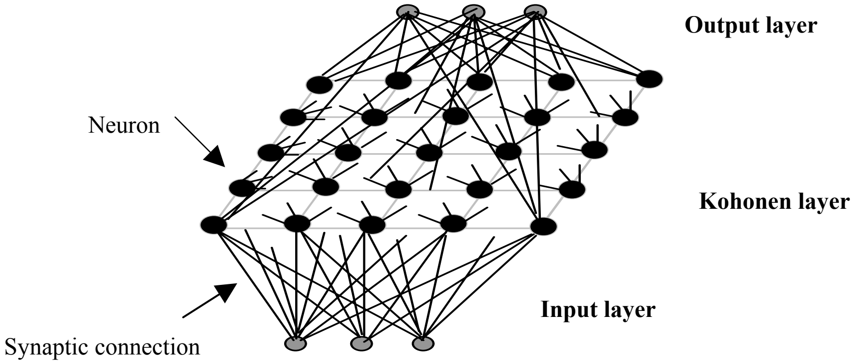


Figure 3-1: Kohonen model

A SOM consists of three layers, namely, input layer, competitive or Kohonen layer, and output layer. Input layer accepts fixed number of input patterns (represented by a vector) from the environment. The competitive layer is the most important layer, where each of the neurons (competitive unit) is tuned to input patterns in an orderly fashion.

The competitive layer comprises of a single layer (organized in 1-dimension, 2-dimensions, or n-dimensions. Typical implementations are 1 or 2-dimensions) of processing neurons. It has two different types of connections: forward connections and lateral connections, as shown in Fig.3.2. Forward connection links the input to the neurons and responsible for updating synaptic weights, and lateral connection establishes a bridge between neurons and responsible for creating computation among them. The winner (the winner takes all neuron) is selected through comparing the distance scores.

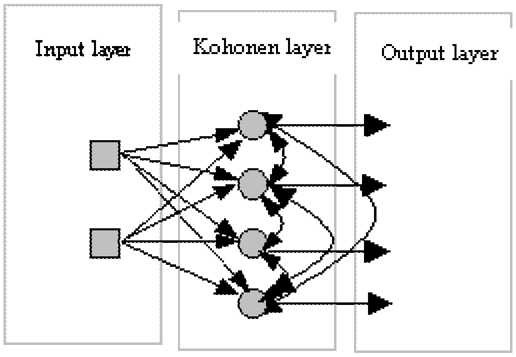


Figure 3-2: Connection in SOM

In the SOM, synaptic weights both of the wining neuron and its neighborhood are allowed to update. Actually, training in the SOM begins with the neighborhood of a fairly large size. As training proceeds, the neighborhood size gradually decreases to the specific value. Depending on application, Kohonen neighborhood has various shapes, such as bubble, rectangular, square, hexagonal, Gaussian, etc.

3.2.1 SOM learning algorithm

Kohonen SOM is trained without any supervision, i.e. learning is based on finding the hidden patterns and extracting features from the data.

Learning procedure has following steps:

- Competitive step
- Cooperative step
- Synaptic modification step
- Convergence step

Competitive step:

Let $X_i = \{x_{1i}, x_{2i}, ..., x_{ni}\}$, a *n*-dimensional input vector, which is fed into the input layer, comprising of *m* number of input neurons. These input neurons are connected with the neurons in the competitive layer. The connection (synaptic) weights of the competitive neuron *j* can be expressed as $W_j = \{w_{j1}, w_{j2}, ..., w_{jn}\}$, which is chosen either randomly (between 0~1) or using some heuristic.

The Euclidean distance (other distance measures such as Hamming distance, Manhatan distance, Hamming distance [27] are also possible choices) between input vector, X_i and weight vector W_j is computed as follows:

$$y_i = \| X_i - W_j \| = \sqrt{\sum_{k=1}^{n} (x_{ik} - w_{jk})^2} \tag{3.1}$$

Using (3.1), distances are computed for all neurons in the competitive layer. Then, the winner-takes-all neuron, j_X is computed using the minimum-distance Euclidean criterion as follows:

$$j_X = \min_{\forall j} \| X_i - W_j \|, \qquad j = 1, 2, ..., m \tag{3.2}$$

Cooperative step:

In this step, the cooperation between wining neuron and neighborhood is established through defining the shape and statistics of the neighborhood around the winning neurons.

Synaptic Modification step:

In this step synaptic weights of wining neurons and its neighborhood are modified according to Hebbian learning rule [26]:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{3.3}$$

$\Delta w_{ij}(t)$, weight correction at the iteration *t* can be written as

$$\Delta w_{ij}(t) = \alpha[x_i - h_{j,i}(t)w_j(t)] \tag{3.4}$$

Where, α , learning rate, is defined as:

$$\alpha(t) = \alpha_0 \exp(-t/\tau) \tag{3.5}$$

Where, α_0 is the initial learning rate (usually 0.1) and τ is the time constant (usually 1000). In (3.4) $h_{j,i}(t)$ is the neighborhood function. For Gaussian distribution, it can be expressed as:

$$h_{j,i}(t) = \exp\left(\frac{-d_{j,i}^2}{2\sigma^2(t)}\right) \tag{3.6}$$

Where d corresponds to the distance between two vectorial locations, and σ corresponds to the width of neighborhood function.

Convergence step:

Following above-mentioned steps, synaptic weights are modified so that feature map between inputs and outputs is formed. For whole input data, training process is repeated until some specific convergence criterion (either threshold value or number of epochs) is met.

3.3 Overview of WordNet

SOM plays an important role in clustering text data. It provides a unique mechanism of clustering, through which a large amount of text data is organized into a small number of meaningful clusters. However, this clustering method cannot bind the semantically close sentences together due to the lack of background knowledge. Based on the experience of other researchers [28,29], it is expected that if the semantically similar sentences could be grouped together, better language models could be made due to the reduced sparseness in data. Based on the assumption, we tried to integrate WordNet in our clustering process.

WordNet [19], an electronic lexical database, has been recognized as an important resource for researchers in human knowledge processing and information retrieval communities. WordNet is sometimes treated as online dictionary or thesaurus. But, it is much more than these. Information in WordNet is organized in hierarchical way, from top to bottom, where top level expresses some sort of abstract concept, and siblings express more specific concepts. Such a way, nouns, verbs, adjectives, and adverbs are organized into synonym groups, called synsets. Each synset is followed by its definition, called gloss.

WordNet supports two types of relations: semantic relation and lexical relation. Semantic relation establishes linkage among hyponyms, hypernyms, meronyms, homonyms, Holonyms, etc, and lexical relation establishes linkage with antonyms[1].

3.4 Data Clustering using SOM

Traditionally data clustering comprises of following steps that are discussed in more detail in the following sub-chapters:

- Data collection
- Preprocessing
- Vector representation, and
- Clustering

3.4.1 Data collection

Data collection is the first step for generating language model. For domain dependent application, data is usually collected from the domain specific resources, and for domain independent application, data could be collected either from a set of domains or from a generic domain.

But, data collected from the above said sources are raw data, which is not ready to use. In order to make the data usable, a lot of pre-processing tasks like removing of unwanted symbols, words, html tags, xml tags, punctuation marks, numeric, and stop words to be done.

In the following subsection, preprocessing step is described.

3.4.2 Preprocessing

Usually, the dictionaries of unwanted symbols, stop words, and common words are generated manually through analyzing huge texts related to the domain of interest. But in practice, this kind of effort is not sufficient, as the collection of text data may contain some less important terms. Finding these terms manually require a great effort. Instead, there are many ways to automate this task by using analysis like kai squire, tf, ifd, tf-idf [30,31], etc.

[1] According to the Ultralingua English Dictionary [43], the terms hypernym, hyponym, meronym, holonym, and antonym are defined as follows:
Hypernym: word that is more generic than a given word; SYN (synonym). superordinate, superordinate word.
Hyponym: A word that is more specific than a given word; SYN. subordinate, subordinate word.
Synonym: Two words that can be interchanged in a context are said to be synonymous relative to that context; SYN. equivalent word.
Meronym: A word that names a part of a given word; brim and crown are meronyms of hat; SYN. part name.
Holonym:. A word that names the whole of which a given word is a part; hat is a holonym for brim and crown; SYN. whole name
Antonym: Two words that express opposing concepts; the antonym of happy is sad; SYN. opposite word, opposite.

3.4.3 Vector representation

Once the data is processed, it is easy to find a list of unique words from it. According to the WEBSOM [32], each text (sentence/paragraph/document) to be mapped onto some representation language. There are many approaches available. Among them one of the most widely used representation languages is single term full text indexing [30]. In this approach, a text, $\xi = \{t_1, t_2, ..., t_l\}$ (where t_l corresponds to *l th* term in a text) is represented by feature vector as:

$$\hat{\xi} = \left(f(\tau_1), f(\tau_2), .., f(\tau_n) \right), \tag{3.7}$$

Where $\tau_i \in \Gamma$, and $\Gamma = [\tau_1, \tau_2, .., \tau_n]$, a list of unique terms extracted from the whole data set. There are many strategies available for specifying value of $f(\tau_i)$, corresponds to the importance of this feature in describing the particular document. The importance is a fuzzy concept. One may represent the importance as a scalar in the range of 0~1, where each of the values within this range is the measure of important-ness of particular feature to describe the document. Increasing value, other than zero is proportional to the increased importance of the feature in question. Other alternative is the binary representation, where importance is represented either by zero or one.

Although text representation using the above strategies is easy to implement, it is impractical for the case where vector size reaches over some thousands. In order to handle this situation, semantic indexing, principle component analysis, etc have been proposed [33]. But, in this thesis, we are going to implement a new simple indexing method that represent the text as a sequence of normalized indices. The fundamental difference between conventional representation and our proposed representation are stated as below:

Conventional representation:

A text is represented as of (3.7), where dimension of the feature vector is fixed, and equal to the total number of unique terms in whole document collection. For the binary single term indexing method, we can rewrite (3.7) as

$$\xi_{con} = \left(f(\tau_1), f(\tau_2), .., f(\tau_n) \right) \tag{3.8}$$

Where $f(\tau_i)$ is:

$$f(\tau_i) = \begin{cases} 1 & \text{if} \quad t_j = \tau_i \\ 0 & \text{otherwise} \end{cases}$$

Proposed representation:

According to our proposed approach, a text is represented as

$$\xi_{pro} = \left(f(t_1), f(t_2), .., f(t_l) \right) \tag{3.9}$$

Where $f(t_i)$ is:

$$f(t_i) = \begin{cases} p^* & \text{if} \quad t_i = \tau_p \\ 0 & \text{otherwise} \end{cases}$$

Where $\tau_p \in \Gamma$, $\Gamma = [\tau_1, \tau_{2,} ..., \tau_p, ..\tau_n]$, and p is the index of the particular term in the list, and p^* is the normalized index (usually $p^* = p/10,000$).

3.4.4 Clustering

Once the vector representation is done, data is ready to be clustered. Clustering using SOM needs to specify some parameters like, cluster size, iteration numbers, learning parameters, and the topology of neighborhood function, etc.

3.5 Integrating WordNet in the Clustering Process

Clustering process using self-organized map tries to map all similar texts into a particular cluster. In this case, similarity is measured as the overall distance between the two texts. When we say that two sentences are more or less similar, that means the constituent words in each sentence are more or less similar. If any word is dissimilar, even though it conveys same meaning, two sentences may not come into particular cluster. So, traditional clustering process is not intelligent enough to grab the inherent semantics of the sentence. In order to grab this information, we need a knowledge source. WordNet is such a resource that can exploit the background knowledge. It provides the more general concepts for most of the terms appearing in the text. And consequently it helps in identifying related topics.

<u>Conventional representation incorporating WordNet</u>:

In order to replace the term, appeared in the text with the concept, we rewrite (3.7) as:

$$\widehat{\xi}_{con} = \left(f(\widehat{f}(\tau_1)), f(\widehat{f}(\tau_2)), .., f(\widehat{f}(\tau_n)) \right) \tag{3.10}$$

Where $\widehat{f}()$ is a mapping function that map the term into concept as:

$$\widehat{f}(\tau_i) = \begin{cases} \delta_i & \text{if } \delta_i \text{ exists in wordNet} \\ \tau_i & \text{otherwise} \end{cases} \tag{3.11}$$

Here, δ_i corresponds to the concept of term τ_i.

Using (3.11), create a modified list Γ' as:

$\Gamma' = [\tau_1', \tau_2', .., \tau_n']$, Where, τ_i' is either δ_i or τ_i based on (3.11)

Now, $f(\widehat{f}(\tau_i))$ is expressed as:

$$f(\widehat{f}(\tau_i)) = \begin{cases} 1 & \text{if } \widehat{f}(t_j) = \tau_i' \\ 0 & \text{otherwise} \end{cases}$$

Proposed representation incorporating WordNet:

In this approach, text is represented as:

$$\widehat{\xi}_{pro} = \left(f(\widehat{f}(t_1)), f(\widehat{f}(t_2)),.., f(\widehat{f}(t_l)) \right) \tag{3.12}$$

Where $\widehat{f}()$ is a mapping function, that map the term into concept as:

$$\widehat{f}(t_i) = \begin{cases} \delta_i & \text{if } \delta_i \text{ exists in WordNet} \\ t_i & \text{otherwise} \end{cases} \tag{3.13}$$

Now, $f(\widehat{f}(t_i))$ is expressed as:

$$f(\widehat{f}(t_i)) = \begin{cases} p^* & \text{if } \widehat{f}(t_i) = \tau_p' \\ 0 & \text{otherwise} \end{cases}$$

Where $\tau_p' \in \Gamma'$, $\Gamma' = [\tau_1', \tau_2', .\tau_p'., \tau_n']$, and p^* is the normalized index.

Since assignment of terms to concepts in WordNet is ambiguous, replacing terms by concepts may add noise to the representation and may induce a loss of information. In order to overcome this problem, various word-sense disambiguation techniques [34,35,36] have been studied and applied by the researchers in the wide area of NLP and NLU, but none of them provide practical solution to this problem. In our experiment, a simple heuristic (most frequent concept) is used, which is claimed to disambiguate around 70% sense [35].

3.6 Generation of Topical Language Models

Statistical language model generation is usually performed by using some existing tools like CMU [21] and HTK [37]. In our research CMU toolkit is used. Fig. 3.3 depicts the generation of topical language models:

Using WEBSOM tool [32], a large preprocessed text data (BNC) is segmented into some pre-defined number of clusters. Since, each of the clusters can be treated as a collection of data, representing some topic (because of the properties of SOM), topical language models can be generated out of these cluster specific data using CMU language model toolkit [21].

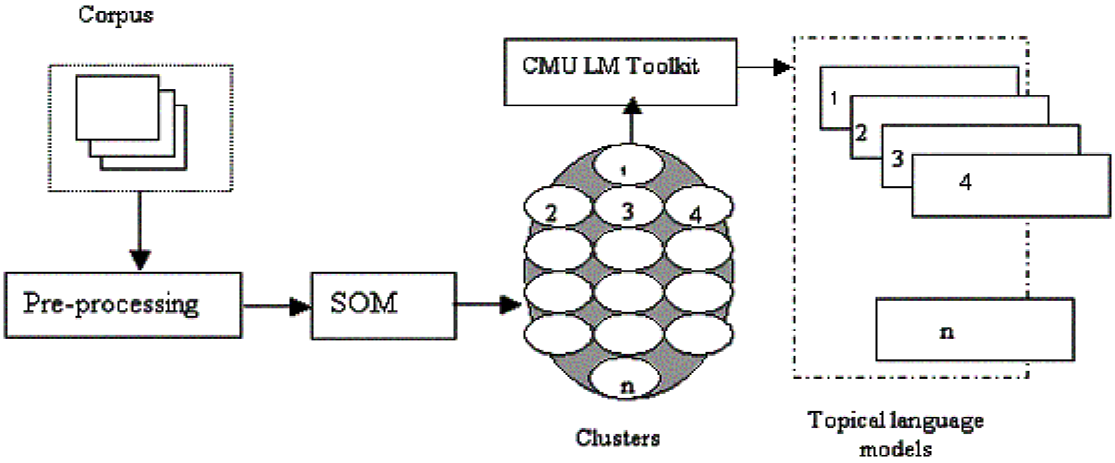


Figure 3-3: Generation of topical language models

Chapter 4

Task Dependent Models and its Applications

CMU Speech decoder comprises of three important components: lexical model or dictionary, language model, and acoustic model. The dictionary provides phonetic transcription for each of the vocabulary words. Language model provides uni-gram, bi-gram, and tri-gram probabilities and back-off probabilities for uni-gram and bi-gram events. Acoustic model provides HMM parameters for each of the phonemes.

Like other versions of CMU decoder, Sphinx3 provides built-in language model and acoustic models, which are usually used for building a generic type of application. But, for building a domain dependant small application, language model is usually re-built with domain specific corpus using LM toolkit or algorithm like *quickLM* [38].

Conventionally, acoustic model, which comes in bundle with the decoder, is used without any modification. For any application, irrespective of small or large, system always loads this whole bundle and occupies unnecessary huge memory space. Actually, this is one of the crucial problems for macro device like cell-phone, PDA and etc. In order to save the memory space and boost up the loading speed, it is possible to extract the necessary part of the acoustic model. This chapter focuses on this particular issue along with a promising solution of generating a language model from a small corpus.

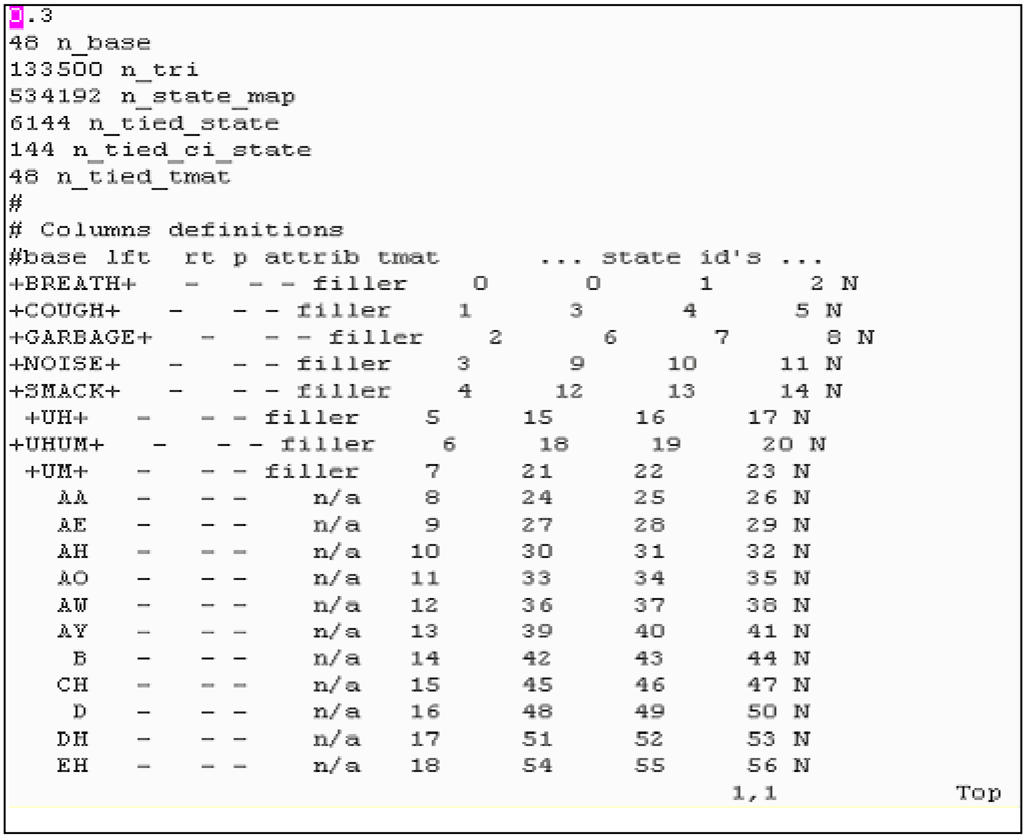
4.1 Extraction of Task-dependent Acoustic Model Parameters

Sphinx3, acoustic model comprises of a collection of five files [39] namely:

1) Model definition file: The model definition file defines the set of base-phone and tri-phone HMMs. It maps each HMM state to a senone (tied state), and each HMM to a state transition matrix. Fig. 4.1 shows the snapshot of model definition file.
2) Gaussian mean and variance files: The Gaussian means and corresponding variance parameters are separated into the two files. Each of the files contains all the Gaussian codebooks. Fig. 4.2 shows the snapshot of Gaussian mean or variance file.
3) Mixture weights file: The mixture weights file contains the Gaussian mixture parameters for all the senones. The snapshot of it is shown in Fig. 4.3.

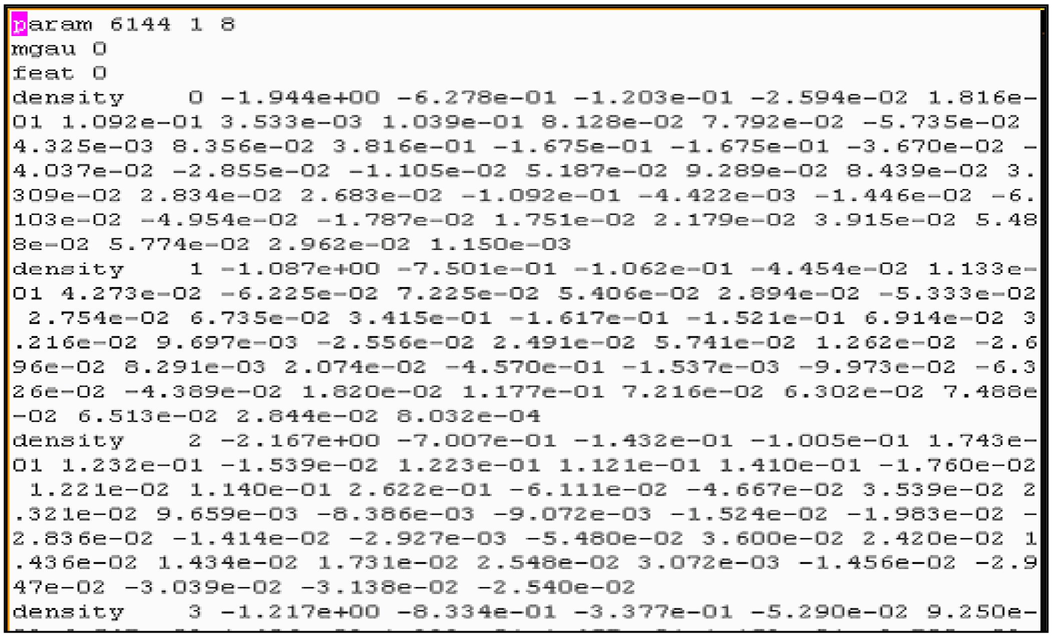
4) State transition matrix file: The state transition file contains all the HMM state transition probabilities in the model.

5) Sub-vector quantized model file: sub-vector quantized model file is an optional file that contains an approximation of the acoustic model



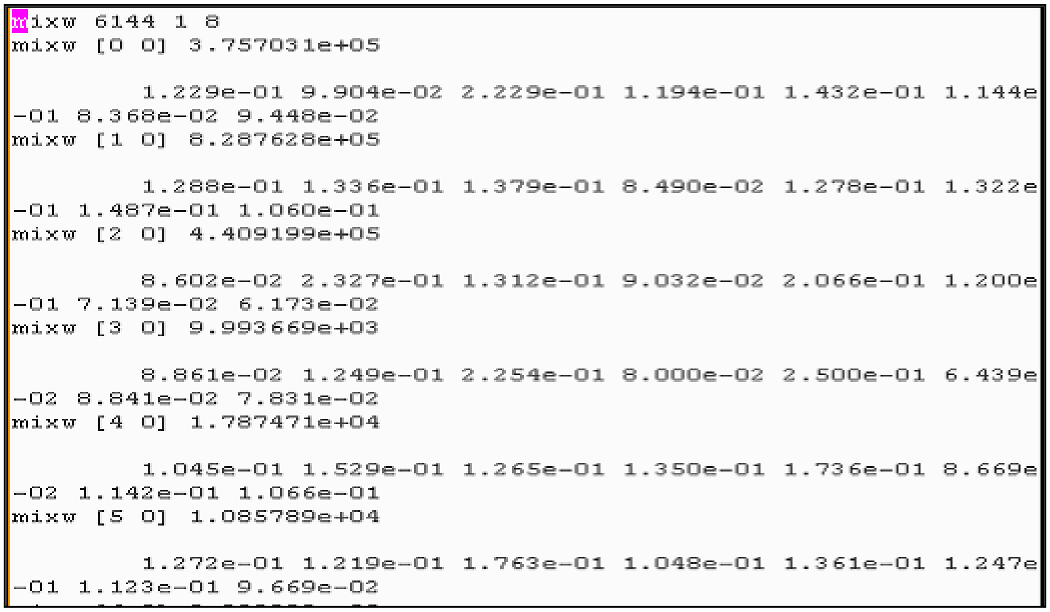
```
0.3
48 n_base
133500 n_tri
534192 n_state_map
6144 n_tied_state
144 n_tied_ci_state
48 n_tied_tmat
#
# Columns definitions
#base lft  rt p attrib tmat       ... state id's ...
+BREATH+   -   - - filler    0       0       1       2 N
+COUGH+   -   - - filler    1       3       4       5 N
+GARBAGE+   -   - - filler    2       6       7       8 N
+NOISE+   -   - - filler    3       9      10      11 N
+SMACK+   -   - - filler    4      12      13      14 N
 +UH+   -   - - filler    5      15      16      17 N
+UHUM+   -   - - filler    6      18      19      20 N
 +UM+   -   - - filler    7      21      22      23 N
   AA   -   - -    n/a    8      24      25      26 N
   AE   -   - -    n/a    9      27      28      29 N
   AH   -   - -    n/a   10      30      31      32 N
   AO   -   - -    n/a   11      33      34      35 N
   AW   -   - -    n/a   12      36      37      38 N
   AY   -   - -    n/a   13      39      40      41 N
    B   -   - -    n/a   14      42      43      44 N
   CH   -   - -    n/a   15      45      46      47 N
    D   -   - -    n/a   16      48      49      50 N
   DH   -   - -    n/a   17      51      52      53 N
   EH   -   - -    n/a   18      54      55      56 N
                                               1,1            Top
```

Figure 4-1: A snippet of model definition file



```
param 6144 1 8
mgau 0
feat 0
density     0 -1.944e+00 -6.278e-01 -1.203e-01 -2.594e-02 1.816e-
01 1.092e-01 3.533e-03 1.039e-01 8.128e-02 7.792e-02 -5.735e-02
4.325e-03 8.356e-02 3.816e-01 -1.675e-01 -1.675e-01 -3.670e-02 -
4.037e-02 -2.855e-02 -1.105e-02 5.187e-02 9.289e-02 8.439e-02 3.
309e-02 2.834e-02 2.683e-02 -1.092e-01 -4.422e-03 -1.446e-02 -6.
103e-02 -4.954e-02 -1.787e-02 1.751e-02 2.179e-02 3.915e-02 5.48
8e-02 5.774e-02 2.962e-02 1.150e-03
density     1 -1.087e+00 -7.501e-01 -1.062e-01 -4.454e-02 1.133e-
01 4.273e-02 -6.225e-02 7.225e-02 5.406e-02 2.894e-02 -5.333e-02
 2.754e-02 6.735e-02 3.415e-01 -1.617e-01 -1.521e-01 6.914e-02 3
.216e-02 9.697e-03 -2.556e-02 2.491e-02 5.741e-02 1.262e-02 -2.6
96e-02 8.291e-03 2.074e-02 -4.570e-01 -1.537e-03 -9.973e-02 -6.3
26e-02 -4.389e-02 1.820e-02 1.177e-01 7.216e-02 6.302e-02 7.488e
-02 6.513e-02 2.844e-02 8.032e-04
density     2 -2.167e+00 -7.007e-01 -1.432e-01 -1.005e-01 1.743e-
01 1.232e-01 -1.539e-02 1.223e-01 1.121e-01 1.410e-01 -1.760e-02
 1.221e-02 1.140e-01 2.622e-01 -6.111e-02 -4.667e-02 3.539e-02 2
.321e-02 9.659e-03 -8.386e-03 -9.072e-03 -1.524e-02 -1.983e-02 -
2.836e-02 -1.414e-02 -2.927e-03 -5.480e-02 3.600e-02 2.420e-02 1
.436e-02 1.434e-02 1.731e-02 2.548e-02 3.072e-03 -1.456e-02 -2.9
47e-02 -3.039e-02 -3.138e-02 -2.540e-02
density     3 -1.217e+00 -8.334e-01 -3.377e-01 -5.290e-02 9.250e-
```

Figure 4-2: A snippet of Gaussian mean/variance file



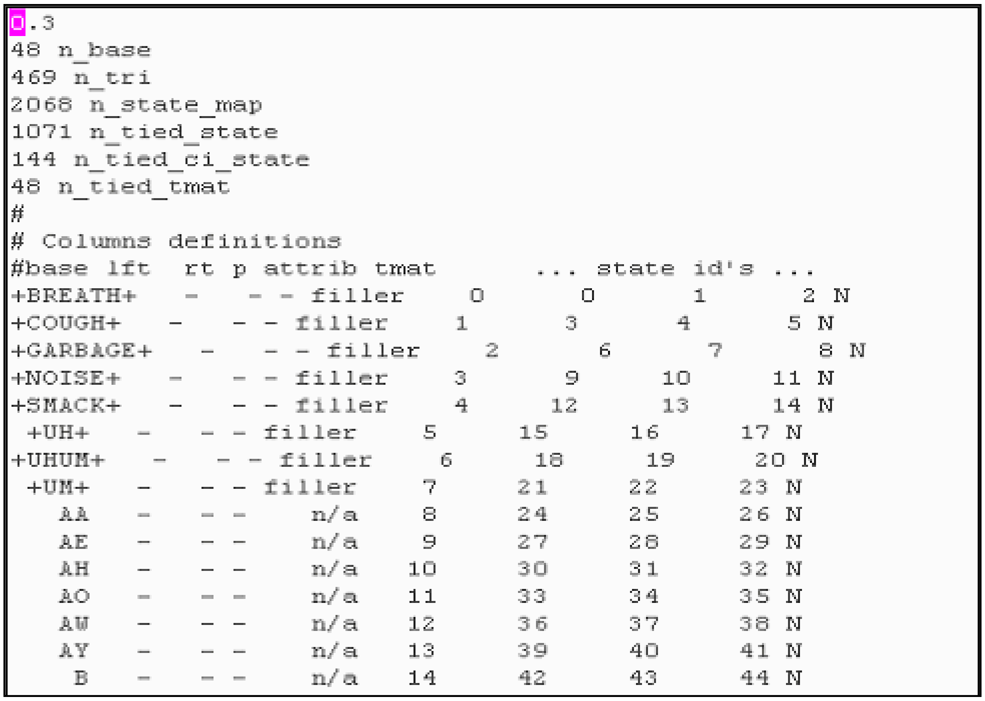
```
mixw 6144 1 8
mixw [0 0] 3.757031e+05

        1.229e-01 9.904e-02 2.229e-01 1.194e-01 1.432e-01 1.144e
-01 8.368e-02 9.448e-02
mixw [1 0] 8.287628e+05

        1.288e-01 1.336e-01 1.379e-01 8.490e-02 1.278e-01 1.322e
-01 1.487e-01 1.060e-01
mixw [2 0] 4.409199e+05

        8.602e-02 2.327e-01 1.312e-01 9.032e-02 2.066e-01 1.200e
-01 7.139e-02 6.173e-02
mixw [3 0] 9.993669e+03

        8.861e-02 1.249e-01 2.254e-01 8.000e-02 2.500e-01 6.439e
-02 8.841e-02 7.831e-02
mixw [4 0] 1.787471e+04

        1.045e-01 1.529e-01 1.265e-01 1.350e-01 1.736e-01 8.669e
-02 1.142e-01 1.066e-01
mixw [5 0] 1.085789e+04

        1.272e-01 1.219e-01 1.763e-01 1.048e-01 1.361e-01 1.247e
-01 1.123e-01 9.669e-02
```

Figure 4-3: A snippet of mixture Gaussian file

Among the above stated five parameters, state transition matrix is constant for any task, and if the task is small (i.e., number of senone is less than 4096), *sub-vector quantized* file is not required.

Now we like to discuss the issue of extracting other three parameters for the task-dependent operation.

Let us imagine a scenario. We have 20 topical language models as well as dictionaries. For a test utterance, we can collect 20 decoded results through running ASR engine, loaded with 20 different topical language model. Now, we like to generate a dynamic language model and a dictionary out of the 20 results. Since the dictionary contains only few pronunciations, context phone (left and right) are limited, and consequently, the number of senones are also limited. For this particular example, we have seen the number of senones required is only 1071, instead of 6144 (provided by Sphinx3). Fig.4.4 shows the snippet of the model definition file for this example.



```
0.3
48 n_base
469 n_tri
2068 n_state_map
1071 n_tied_state
144 n_tied_ci_state
48 n_tied_tmat
#
# Columns definitions
#base lft  rt p attrib tmat      ... state id's ...
+BREATH+   -   - - filler    0      0      1      2 N
+COUGH+   -   - - filler    1      3      4      5 N
+GARBAGE+   -   - - filler    2      6      7      8 N
+NOISE+   -   - - filler    3      9     10     11 N
+SMACK+   -   - - filler    4     12     13     14 N
 +UH+   -   - - filler    5     15     16     17 N
+UHUM+   -   - - filler    6     18     19     20 N
 +UM+   -   - - filler    7     21     22     23 N
   AA   -   - -    n/a    8     24     25     26 N
   AE   -   - -    n/a    9     27     28     29 N
   AH   -   - -    n/a   10     30     31     32 N
   AO   -   - -    n/a   11     33     34     35 N
   AW   -   - -    n/a   12     36     37     38 N
   AY   -   - -    n/a   13     39     40     41 N
    B   -   - -    n/a   14     42     43     44 N
```

Figure 4-4: A snippet of task-dependent model definition file

Now, if we analyze the model definition file, shown in Fig. 4.4, we will get an idea of memory requirement for both of the task-dependant and task-independent operations.

For the task-dependant operation:

- N_base, denotes total number of base phones : 48
- N_tri denotes total number of tri-phones : 469
- Each of the model has 4 states, hence total number of states = (48+469)*4 = 2068
- N_tied_state denotes total number of senones = 1071

- Each of the senones has following components:
    - 8 Gaussian means/state
    - 8 Gaussian variances/state
    - 8 Gaussian mixtures/state

Total number of components for 1071 senones = 1071(8*39) + 1071(8*39) + 1071*8 = 676872

Total memory required = 676872 *4 bytes ≈ 2.7MB

For the task-independent operation:

The number of senones = 6144

Total memory required = 6144(8*39) + 6144(8*39) + 6144*8 ≈ 3.9MB

So, model extraction method is able to save a huge memory space for small vocabulary speech recognition system.

4.2 Language Model Generation with Small Corpus

Conventionally, Sphinx3 language model toolkit uses Good-Turing back-off technique for generating a language model. In practice, this technique provides a good solution for data sparseness. CMU developed another algorithm, *quickLM* that is claimed to provide a good language model especially for the small corpora. The *quickLM* uses proportional discounting technique, where a portion of probability mass is discounted from the observed events and redistributed among the unseen events. Empirically, we have found that when vocabulary size is very small (< 100 words), proportional discounting cannot provide enough smoothness in language model. Instead, if we assign same count for each of the events within each of the events (usually uni-gram events are smoothed through assigning average number of counts of all of the uni-grams, bi-gram events are smoothed through assigning average number of counts of all of the bi-grams, and so on), speech recognition performance improves dramatically.

So, in our experiment, we used modified *quickLM* for generating dynamic language model.

Chapter 5
Implementation and Experimental Results

This chapter presents an implementation of a statistical topical language models in large vocabulary continuous speech recognition system. Here, detail about working components and its configuration are discussed. A set of experimental results is also presented to demonstrate the effectiveness of the topical language models.

5.1 Overview of the System

The system going to be implemented has three phases, viz., phase-I, phase-ii, and phase-iii.

1. Phase-i
    a. Data collection
    b. Preprocessing
    c. Encoding with WordNet
    d. Encoding without WordNet
2. Phase-ii
    a. Clustering
3. Phase-iii
    a. Language model generation
    b. Dictionary generation
    c. Speech decoding using ASR engine

The system flow is shown in Fig.5.1.

5.1.1 Data collection

As the first step of this application, we started collecting data, which was BNC [40] (British national Corpus). It contains 4054 texts (including SGML markup) and occupies 1.5Gb of memory. In total, it comprises of approximately 100 million words and about 6 millions sentences. It covers most of the common domains including applied sciences, arts, beliefs and thoughts, commerce and finance, imaginative, leisure, natural and pure science, social science, and world affairs. Tab. 5.1 shows the distribution of BNC corpus.

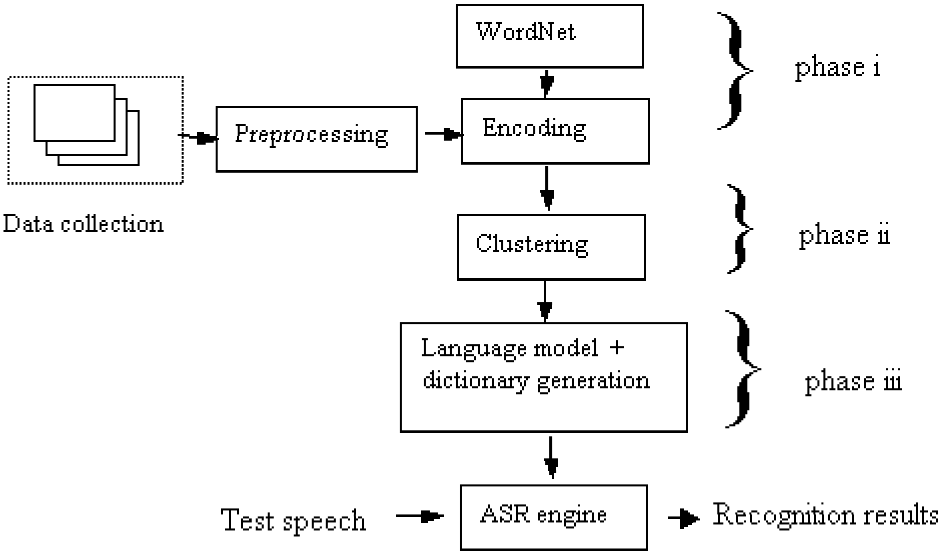


Figure 5-1: System flow diagram

Table 5.1: Domain information and text distribution in BNC

Domain	Texts	Words	%	Sentences	%
Applied Science	370	7104635	8.14	14 357067	7.12
Arts	261	6520634	7.47	321442	6.41
Belief and thought	146	3007244	3.44	151418	3.01
Commerce and finance	295	7257542	8.31	382717	7.63
Imaginative	477	16377726	18.76	1356458	27.05
Leisure	438	12187946	13.96	760722	15.17
Natural and pure science	146	3784273	4.33	183466	3.65
Social science	527	13906182	15.93	700122	13.96
World affairs	484	17132023	19.62	800560	15.96

5.1.2 Preprocessing

BNC is a SGML formatted corpus. It contains a lot of unwanted and noisy words as well as sentences. So, in order to use BNC for clustering, it needs some sort of rectification, which is shown in the following diagram (Fig. 5.2).

We first removed SGML tags, and filtered out all of the non-words, punctuations, and abbreviations and numerals. Subsequently, we got 5,845,344 sentences containing 133,099 unique words. Computing frequency (sentences having more than 20 words), we further reduced the wordlist as 49,233.

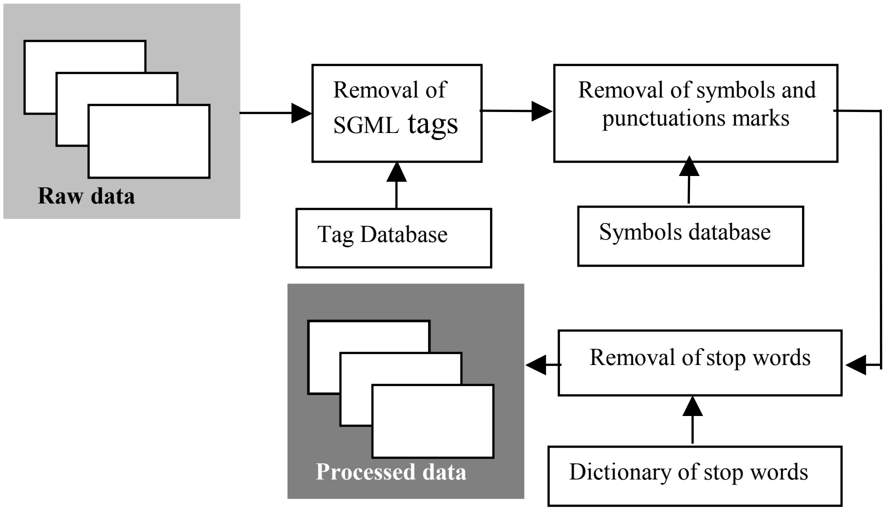


Figure 5-2: Preprocessing scheme

5.1.3 Encoding

Each of the sentences in BNC was encoded using the strategies described in previous section. First all of the sentences were encoded without the involvement of WordNet, and then using WordNet. Each of the sentences was represented by vector of fixed length as 60. In case of a sentence size was less than 60, the rest of the components was filled with 0's, and for the sentence whose size was above 60, just truncated to 60. Note that dimension of the vector was selected through analyzing the vast amount of text data of BNC corpus.

5.1.4 Clustering

In order to group preprocessed BNC corpus into predefined number of clusters, we used WEBSOM toolkit for sentence level clustering with the following specification:

- Initial neighborhood radius was set as 15
- Number of clusters that SOM would produce was set as 5,10,20,20, and 40.
- Hexagonal topology was used in the co-operative step of SOM algorithm.
- The neighborhood function type *bubble* was used.
- Running length (number of iterations) in training was chosen to be 30000 after a number of trials, shown in Fig. 5.3.
- For all other parameters, default values were used.

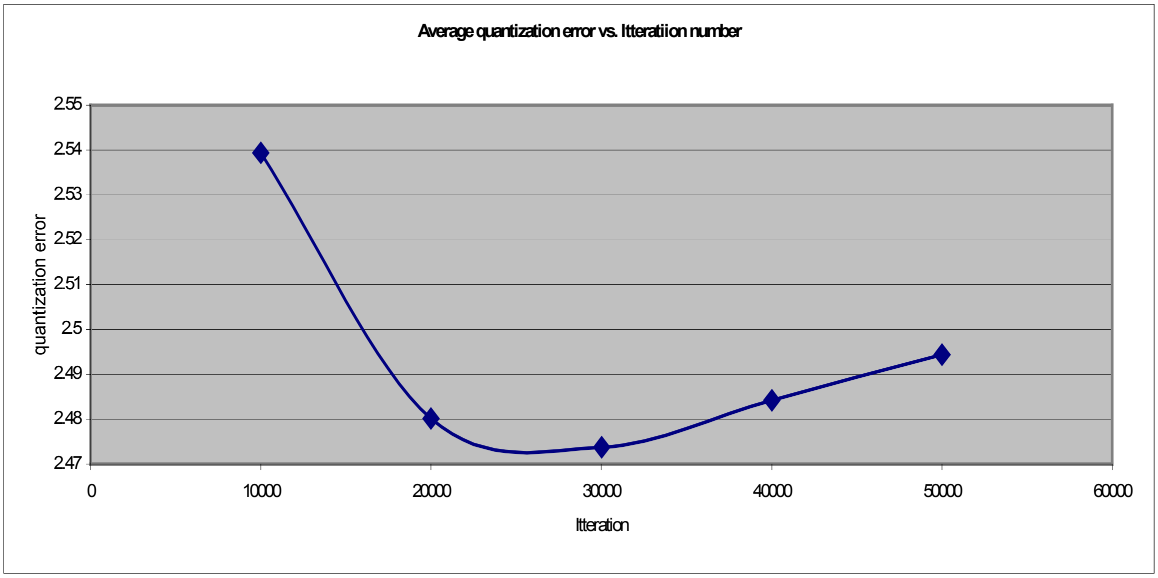


Figure 5-3: Average quantization error vs. iteration

5.1.5 Language model and dictionary generation

The ASR engine has two essential components: Language model, and dictionary. First a big language model was generated from whole pre-processed BNC corpus through using CMU language model toolkit. Then a dictionary was generated with all unique words in the corpus using CMU dictionary generation tool [41].

Same way, each of the topical language models was generated through collecting cluster specific sentences and dictionary with cluster specific unique words.

5.1.6 Speech data collection

In order to study the performance of ASR system, speech data is required. In our experiment, we collected a set of speech data from the VOA website [42]. We first downloaded five audio webcastes (MP3 format) covering health, science, agriculture, development, and explosion. After editing, we got 279 audio files (transcripts are shown in Tab. 5.2~Tab.5.6), which were converted and stored as raw format for subsequent speech recognition experiment.

Table 5.2: Transcription of speech corpus (Subject: Health)

1	earlier research led by professor merill showed that such molecules in milk can suppress the formation of growths
2	but he says this is the first study to show that similar molecules in plants can also suppress cancer
3	the study found that a molecule known as soy glccer reduced the formation and growth of tumor cells in mice
4	some of the mice were born with a gene that leads to colon cancer
5	others were given a chemical that causes the disease
6	the soy glccer passed through the stomach and intestines
7	but professor merrill says it stayed strong enough to suppress cancerous cells in the colon
8	part of the large intestine
9	the next step is to see if the molecule works the same way in humans
10	interest in soy has led to many more food and health products that contain it
11	foods made from soybeans are increasingly popular
12	these are especially popular with older women
13	their bodies no longer produce the female hormone estrogen
14	so they worry about their risk of breast cancer
15	soy contains two substances that are similar to estrogen
16	however experts say one of these might increase the risk of breast cancer in some women
17	they say more research is needed on the different chemicals in soy and the safety of taking them in large amounts
18	earlier this year scientists reported that soy may help men prevent prostate cancer
19	but some men apparently are concerned about the estrogen-like effect of soy
20	so in a different study scientists had men eat much larger amounts of soy than they would normally get in food
21	there were a few side effects reported including breast enlargement
22	and not just because of the taste
23	but researchers at the university of north carolina at chapel hill said none of these effects were serious
24	this voa special english health report
25	studies has found that soy can be good for the health in different ways
26	now research in the united states shows that a molecule in soy may help prevent colon cancer

27	the journal of nutrition published the study by researchers at georgia tech emory university and the karmanos cancer institute
28	al merrill of georgia tech says that soy is known to suppress cancer
29	he says that some of this effect may be from a group of molecules
30	these are called sphingolipids
31	plants and animals have many different kinds

Table 5.3: Transcription of speech corpus (Subject: Science)

1	one of the new studies took place at duke university medical center in durham north carolina
2	the robert c atkins foundation paid for the study but was not involved in the research
3	this organization works to get more people to follow the doctor's ideas
4	one-hundred-twenty overweight adults took part in the study
5	they were between the ages of eighteen and sixty-five
6	they followed either the atkins plan or a low-fat diet for one year
7	after six months the people on the atkins diet lost an average of eleven kilograms
8	those on the low-fat diet lost an average of six kilograms
9	the veterans affairs medical center in philadelphia pennsylvania did the second study
10	this study did not involve the atkins foundation
11	many people who try to lose weight know that no diet is perfect
12	one-hundred-thirty-two adults took part
13	most had diabetes
14	the researchers put half the people on a low-carb diet
15	the other half followed a low-fat diet
16	after one year the low-carb dieters had lost on average as much as eight kilograms
17	yet the low-fat dieters lost about the same amount
18	what happened?
19	the low-carb dieters lost weight faster in the beginning
20	but the low-fat dieters lost weight throughout the year
21	however the study found that the people with diabetes controlled their blood sugar better with low carbohydrates
22	he told dieters to avoid foods high in starch and sugar
23	the new research also found that triglyceride levels fell more on the low-carb diet than on the low-fat plan
24	triglycerides are fats in the blood that can increase the risk of heart disease
25	levels of so-called good cholesterol also appeared to improve with the low-carb diet
26	higher levels of good cholesterol may reduce the risk of heart disease
27	but levels of bad cholesterol did go up in some people
28	doctor walter willett is a nutrition expert at the harvard school of public health

29	doctor willett wrote a commentary on the two studies
30	in his words we can no longer dismiss very-low carbohydrate diets
31	he says doctor atkins should get credit for his observations that many people can control their weight by greatly reducing carbohydrates
32	but other health experts are not satisfied
33	doctor atkins died last year
34	they want more research done to learn the effects of following the atkins diet for long periods of time
35	they warn that people who eat a lot of fat may give themselves a heart attack
36	and they question how good it is for people not to eat things like fruit
37	the atkins diet and other low-carbohydrate plans have had a big effect on the food industry
38	stores sell lots of new low-carb foods as well as lower carb versions of breads and pastas
39	but supporters of the atkins diet say people should not use it as an excuse to fill themselves with fatty foods
40	they say proteins such as poultry fish beef pork and soy products should be the largest part of what people eat
41	but they say the next largest part should be green vegetables
42	after that the plan calls for smaller amounts of fruits oils nuts cheese and beans
43	he fell on an icy street in new york and suffered a head injury
44	the atkins advice is that the smallest part of what people eat should be whole grain foods such as barley oats and brown rice
45	but to lose weight it says eating should center on protein leafy vegetables and healthy oils
46	last week the new york times reported what it said was apparently the first legal action against the atkins diet
47	a florida man said he suffered a blocked artery from high cholesterol after two years on the diet
48	he asked for twenty-eight-thousand dollars
49	the atkins nutritionals company said the case was part of an effort to scare people into not eating any animal protein
50	low-carbohydrate diets or not more people than ever weigh too much
51	the world health organization says this is a serious problem
52	it says the opposite problem hunger affects about eight-hundred-fifty-million people
53	but more than one-thousand-million are overweight and that just counts adults
54	he was seventy-two years old
55	at least three-hundred-million adults are obese severely overweight
56	health ministers around the world now have a plan called the global strategy on diet physical activity and health
57	they approved it in late may at the meeting of the world health assembly in geneva
58	the plan urges people to eat less saturated fats and trans fatty acids
59	food products often list trans fats under the name partially hydrogenated oil
60	the plan also urges people to eat less salt and sugar and more fruits and

	vegetables
61	it calls for more physical activity
62	and it suggests that governments restrict food advertising especially messages aimed at children
63	the plan took two years to develop
64	the sugar industry and several sugar-producing nations had objected to earlier proposals
65	our body uses carbohydrates for energy
66	they wanted to remove any discussion about limits on sugar
67	some sugar producing nations feared that their farmers would be hurt by the new strategy
68	the director general of the who lee jong-wook praised the strategy as a major success in public health policy
69	he said it will provide countries with a powerful tool to fight diseases caused by obesity
70	health officials say poor diet and lack of exercise are among the leading causes of heart disease diabetes and some cancers
71	they say these kinds of diseases now cause about sixty percent of deaths worldwide
72	in the united states the government estimates that one in three adults is obese
73	but health officials warn that the problem is spreading in developing nations as they gain more wealth
74	and the problem is not just among adults
75	a group called the international obesity task force estimates that one in ten children worldwide is overweight or obese
76	it can also make energy from protein and fat
77	the world health organization estimates that about three-million people a year become infected with this disease
78	about one-million of them die
79	most of the deaths are in africa
80	young children and pregnant women suffer the most
81	now the united nations has given its support to another drug to fight malaria
82	it is a traditional chinese herbal medicine called artemisinin
83	this drug comes from a plant called the sweet wormwood
84	chinese researchers discovered artemisinin more than thirty years ago
85	tests took place in the early nineteen-nineties in vietnam
86	malaria spreads through mosquito bites
87	but proteins generally make people feel more satisfied with less food than carbohydrates do
88	new drugs are needed because the parasites that cause the infection develop resistance
89	health experts hope to prevent resistance to artemisinin by giving the drug in combination with other medicines
90	but experts also warn against the overuse of malaria drugs by people who do not have the disease
91	they say that sick people often mistake influenza or other diseases for malaria

	and take anti-malaria medicine
92	this can add to the problem of drug resistance
93	there are home tests for malaria
94	health experts say greater use of these tests could help make sure people take malaria drugs only when they really need them
95	this is one of the main arguments for a low-carb diet to lose weight

Table 5.4: Transcription of speech corpus (Subject: Agriculture)

1	but unlike the green revolution biotechnology has been supported mainly by private investment
2	businesses are unwilling to share trade secrets with countries that do not recognize their property rights
3	so they develop crops for large markets
4	the un food and agriculture organization says little research has been done on food crops like wheat rice potatoes and cassava
5	an fao report last month expressed concern that biotechnology is not helping developing nations
6	six countries grew ninety-nine percent of all biotech crops last year
7	argentina brazil canada china south africa and the united states
8	almost all these crops have special genes to resist damage by insects or by chemicals used to kill unwanted plants
9	this is steve ember with the voa special english agriculture report
10	the fao says there is little research on biotech plants that could resist crop failure in poor countries or provide extra vitamins
11	director-general jacques diouf says scientists generally agree that foods made from genetically engineered crops are safe to eat
12	but he adds that little is known about their long-term effects
13	he also says there is less scientific agreement on the environmental effects so each product must be carefully observed
14	public opinion is a big issue in the debate
15	opponents say there may be unknown health dangers
16	some poor nations have refused any food aid that contains genetically engineered products
17	yet the industry has had some successes recently
18	last month the european union ended a six-year suspension of approval for new biotech foods
19	and brazil has been moving to let farmers plant genetically engineered soybeans
20	there are sixty-eight million hectares of genetically engineered crops
21	this is about five percent of all cropland in the world and expanding
22	but debate over how best to use this biological technology continues
23	experts compare the rise of biotechnology to the period of change in the nineteen-sixties and seventies
24	the green revolution produced the modern systems and chemicals of agriculture

25	productivity increased in many countries
26	today the united nations and others are calling for a gene revolution
27	experts say the world must find new ways to fight hunger and feed its growing population

Table 5.5: Transcription of speech corpus (Subject: Development)

1	and directions about how to use the drugs have pictures so they are easier to understand
2	one example is in burkina faso
3	high fever is the most important sign of malaria
4	experts say children should be treated within twenty-hour hours after their temperature rises
5	at first a cool wet cloth may help reduce the body temperature
6	but children can die within two days if the malaria becomes severe
7	children must receive the correct amount of medicine
8	and they must take all the medicine they are given
9	this is robert cohen with the voa special english development report
10	mothers and health workers are told to take the child to a medical center
11	if the fever is treated but continues after two days
12	other signs of malaria include sleepiness and feeling sick in the stomach
13	the who says people often take patients to traditional healers to treat another effect of malaria: severe shaking
14	but it says the healers should be trained to tell them they must go to a hospital
15	the world health organization has published a guide in an effort to increase malaria treatment at home
16	this information tells about how to train and educate mothers and other people about malaria
17	in uganda for example communities have elected a person to learn the signs of malaria and provide medicine
18	teachers and store keepers are also trained to help educate the community about malaria
19	the guide is called scaling up home-based management of malaria
20	malaria is estimated to kill another child in africa every thirty seconds
21	internet users can find it at wwww.ho.int
22	again wwww.ho.int
23	but there is new evidence that treatment of malaria at home can save many lives
24	this is called home-based management
25	home-based management is being used in several countries in africa
26	these include uganda ghana and nigeria
27	local health workers and mothers of young children are trained to recognize the signs of malaria
28	they are taught to seek treatment immediately
29	store keepers are trained to sell the right amount of medicine for the age of the patient

Table 5.6: Transcription of speech corpus (Subject: Explosion)

1	its waters flow through deep mountain canyons some of them are more than five-hundred meters deep
2	it continues across great flat plains areas and deserts feeding rich agricultural areas along the way
3	the rio grande flows south to the cities of el paso texas and ciudad juarez in the mexican state of chihuahua
4	then it turns in a southeast direction
5	here it becomes the border line between the united states and mexico for two-thousand kilometers
6	from this point in the most western part of texas the rio grande flows east to where the river empties into the gulf of mexico
7	along its way the river flows through or past the cities of albuquerque and las cruces new mexico by el paso and ciudad juarez
8	the last cities it touches are brownsville texas on one side of the border and matamoros mexico on the other
9	on its long trip to the sea the rio grande expands as a number of rivers flow into it
10	in the united states those rivers include the pecos devils chama and puerco rivers
11	today we tell about one of the most famous rivers in north america the rio grande
12	in some places the river is more than ten meters deep
13	but in many places on the river there is not much water flowing
14	this lack of water is a sign that much of the river is used for growing crops and providing water supplies for the expanding population
15	this is not a new use for the rio grande
16	there is much evidence that the ancestors of the pueblo indians in new mexico used water from the river to grow crops for thousands of years
17	the pueblo ancestors arrived in the southwest of what is now the united states about two-thousand years ago
18	although their food mostly came from hunting they grew some crops for food
19	the pueblo civilization went through a number of changes over time
20	it forms the border between the southwestern state of texas and mexico
21	such as the navajo and apache indians
22	a severe dry period more than six-hundred years ago also affected the pueblo civilization
23	the weather is believed to be one reason some of the great cities of the southwest area were left empty as the pueblo ancestors moved closer to the rio grande
24	a major change for these people began soon after the first europeans came to the rio grande
25	they first were looking for a way to the pacific ocean
26	soon they were more interested in searching for riches such as those captured by spanish explorer hernando cortes
27	in fifteen-twenty-one cortes conquered the great aztec empire in what is modern mexico
28	cortes seized huge amounts of gold and jewels from the aztecs

29 many spanish explorers heard the stories about the wealth of the aztecs

30 they hoped to find similar wealth among other indian groups in north America

31 the rio grande been has important in the history and development of the united states and mexico

32 some explorers hoped that the rio grande would lead them to indian nations that also possessed gold and jewels

33 the most famous explorer of the rio grande territory was francisco vazquez de coronado

34 he arrived at the rio grande in fifteen-forty

35 earlier explorers of the rio grande area said they had heard of great indian cities on a river in the north

36 the stories they heard were about cities that had treasures of costly stones such as turquoise and emeralds

37 the spanish explorers also believed there was gold silver iron and copper in the mountains to the north

38 spain had already taken great wealth from the incas of peru and the aztecs of mexico

39 why not also take the riches of the indians cities north of mexico?

40 so the spanish viceroy of mexico gave an order which would change the history of north america

41 he asked coronado to lead an army of spanish soldiers to the north

42 however the river has a different name in mexico

43 they were ordered to conquer new land for the king of spain

44 land that the spaniards called cibola

45 coronado and his soldiers did not find the cities of gold that they were seeking

46 instead they found many indian towns with tall houses and rich fields full of corn and other plants

47 the people were peaceful farmers

48 they did not remain peaceful

49 the spanish soldiers did things to the pueblo indians that made them angry

50 so the indians decided to push the spaniards out of their land

51 the spanish soldiers won the battles with the pueblo indians and destroyed many of their towns

52 then the spanish searched for gold and silver

53 it is called rio bravo del norte

54 they found none

55 they returned to mexico with nothing to show for their struggles in the areas of the rio grande river

56 coronado died in mexico city in fifteen-fifty-four

57 he was forty-four years old

58 again the spanish tried to establish a colony in the area

59 they tried four times and failed each time

60 in fifteen-ninety-eight a large spanish army marched north from mexico

61 the king of spain ordered that a colony be established on the river north of mexico

62 the name of the new colony was to be new mexico

63	traveling with this army were many families roman catholic priests and thousands of cattle
64	the rio grande begins its three-thousand kilometer trip to the gulf of mexico high in the rocky mountains in the state of colorado
65	they established a colony on the river where some pueblo indians already lived
66	the spanish called it san juan
67	the indians seemed to accept them
68	but the peace did not last
69	suffering and tragedy spread through the land as the spanish and indians fought
70	the spanish priests and the settlers in san juan began to protest against the cruel treatment of the indians
71	it would be better they said not to have any spanish colony in new mexico than to built one on such crimes against the native peoples
72	finally in sixteen-six the king of spain ordered the end of the colony at san juan
73	the spanish settlers left but the indians remained at what is now san juan pueblo
74	the spanish would be back
75	it begins almost four-thousand meters up where the river is fed by melting snow
76	in sixteen-ten a new governor of new mexico arrived
77	a new capital was built called santa fe
78	it still is the capital
79	this time the goal of the spanish government was to spread the christian religion among the indians
80	the brothers of the order of saint francis were not like the earlier spaniards
81	at first the indians resisted them
82	but over time they understood that these men did not want to oppress them
83	the franciscans wanted to teach the indians about jesus christ
84	the franciscans helped the pueblo indians build many beautiful churches throughout the area
85	the churches were built with local materials
86	soon other small streams flow into the river
87	they did not look like the traditional churches of europe
88	some of these churches still stand today
89	they are very popular with artists
90	the franciscans wanted the indians to be protected
91	the indians were not sure who they should obey
92	while this dispute was taking place there was a long dry period that caused people in the area to starve
93	then the disease smallpox began taking the lives of many indians and spanish settlers
94	there was a violent rebellion by the pueblo indians and the spanish were forced to leave the rio grande area
95	yet they were not to be pushed out for long
96	increasing its size as it flows generally south through the state of new mexico

5.1.7 Speech decoding

In order to justify the effectiveness of topical language models over big language model, a set of experiments have been performed. First, a set of experiments has been performed using the topical models generated without the assistance of WordNet (labeled as *without WordNet*), and then with WordNet (labeled as *with WordNet*). The experimental conditions and results are presented in the following sub-sections.

5.1.7.1 Experiment #1: Finding optimum number of topical models

In order to decide the optimum number of topical language models (in other word, optimum number of clusters), an experiment has been performed under following condition:

Experimental condition:

- No. of topical models: 5,10,20,30, and 40
- Recognition engine: Sphnix3
- Test utterance: randomly selected 100 VOA utterances, collected from various domains.

Tab.5.7 shows the recognition performance

Table 5.7: Recognition performance for experiment #1

Number of Clusters	Average Recognition Performance
5	75.3%
10	75.4%
20	78.0%
30	78.7%
40	78.0%

Results show the gradual improvement in performance with the increased number of clusters, but after 20 (cluster's quantity), the performance does not change significantly, rather it starts degrading. Hence, our subsequent experiments have been done using 20 clusters.

5.1.7.2 Experiment #2: Find the effectiveness of topical language models over big language model

In order to justify the effectiveness of topical language models over big language model, a set of experiments has been performed under following condition:

Experimental condition:

- No. of topical models: 20
- Base model: big language model
- Recognition engine: Sphnix3
- Test utterance: 279 VOA utterances (Tab.5-2~ Tab.5.6)

The results are shown in Fig.5.4 ~Fig.5.6.

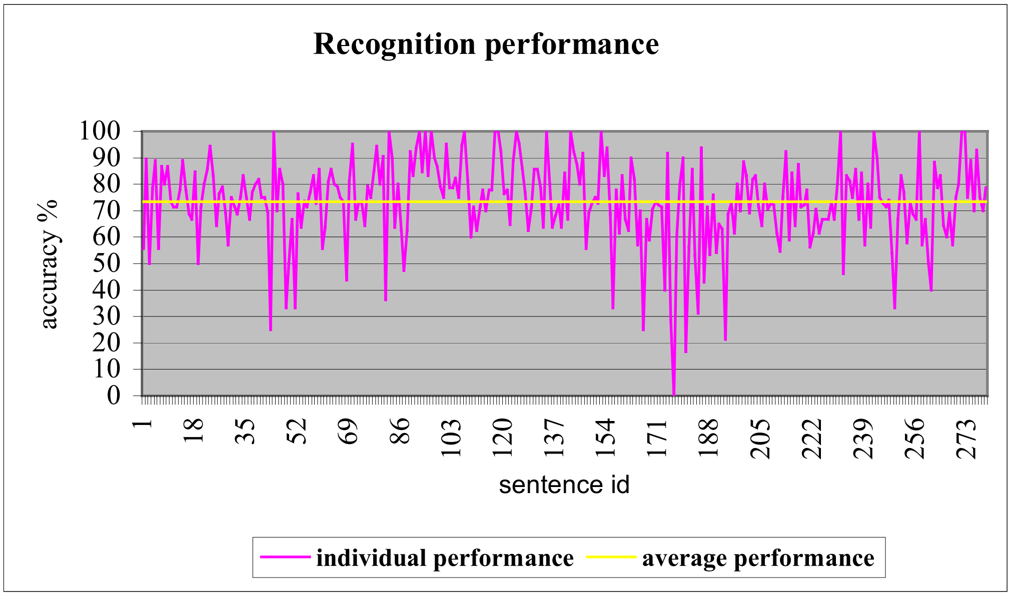


Figure 5-4: Recognition performance using big language model

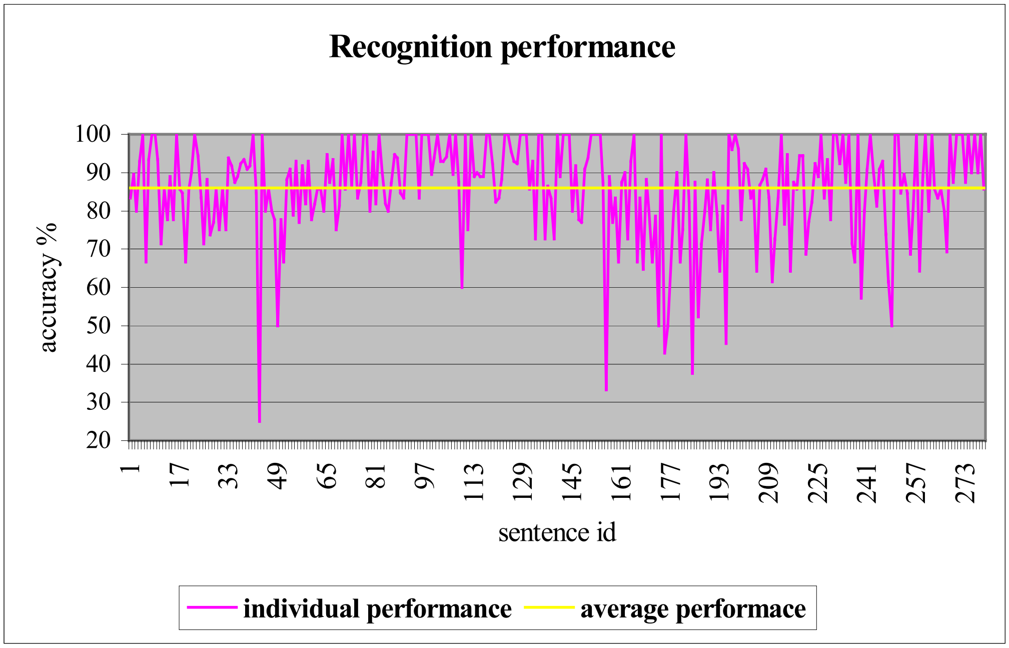


Figure 5-5: Recognition performance using topical language models (without WordNet)

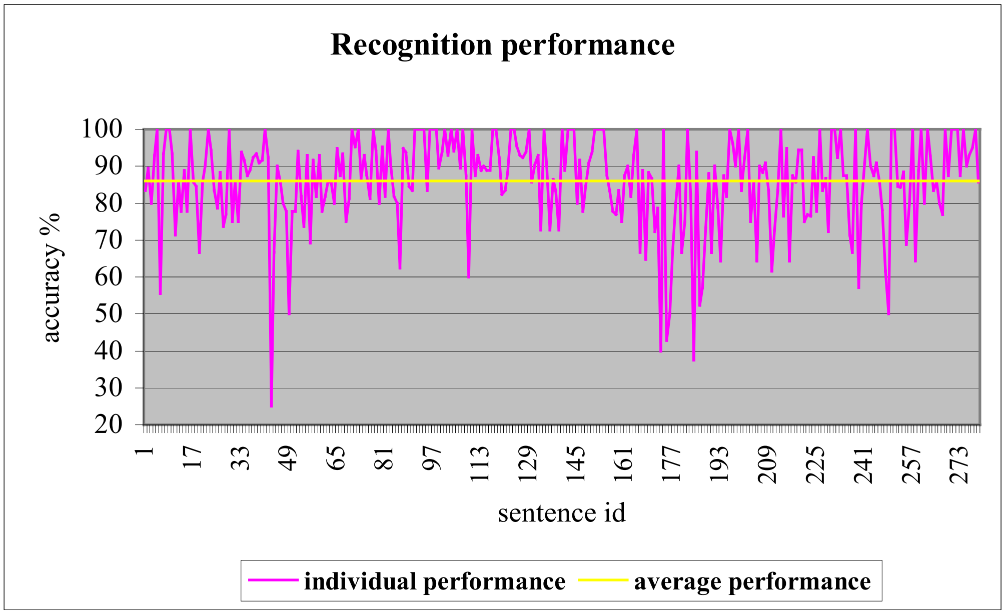


Figure 5-6: Recognition performance using topical language models (with WordNet)

Results show the better performance of topical language models compared with the performance of big language model. It also shows a marginal improvement in recognition performance using WordNet.

5.1.7.3 Experiment #3: Strategy for enhancing the recognition performance using topical language models

Although topical language models perform better than the big language model, the task of finding the best performing model has not yet been solved. Since, our goal is to enhance the recognition performance, we adopted following strategies:

Strategy # 1:

- Collect recognition results for a test utterance from the speech decoder loaded with 20 different topical language models (note: number of topical language model is 20).
- Make a dynamic language model out of the 20 results (using same language model generation tool).
- Feed the same utterance once again in the input of the decoder loaded with a dynamic language model.
- Verify the results

Strategy # 2:

- Collect recognition results for a test utterance from the speech decoder loaded with different topical language models.
- Convert each of the results as a string of connected words:

  Example:

  ASR output:

  I am going to test the performance of ASR system using my strategy.

  Processed output after connecting words:

  I_am_going_to_test_the_performance_of_ASR_system_using_my_ strategy.
- Make a dynamic language model out of the 20 results.
- Feed the same utterance once again in the input of the decoder loaded with a dynamic language model
- Verify the results

Following the above strategies, a set of experiments has been performed using WordNet and without WordNet. The results are shown in Fig.5.7~Fig.5.10.

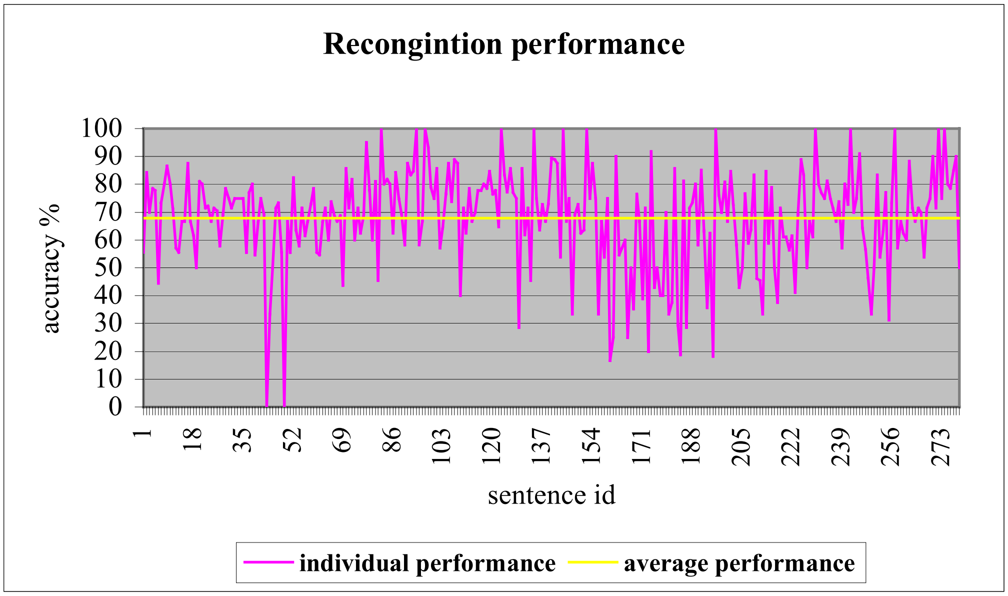


Figure 5-7: Recognition performance using dynamic language model (Strategy #1: without WordNet)

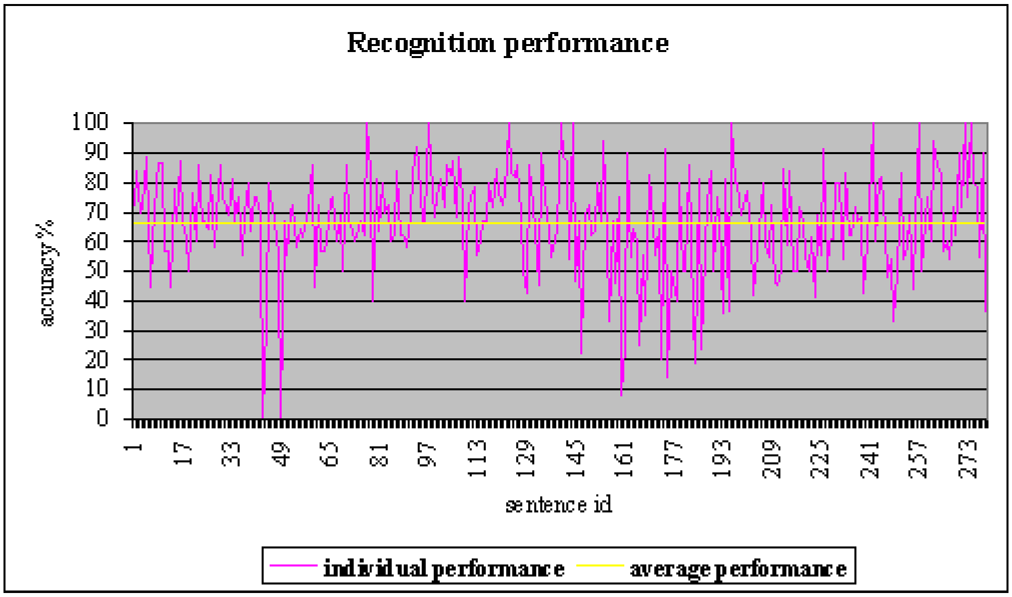


Figure 5-8: Recognition performance using dynamic language model (Strategy #1:with WordNet)

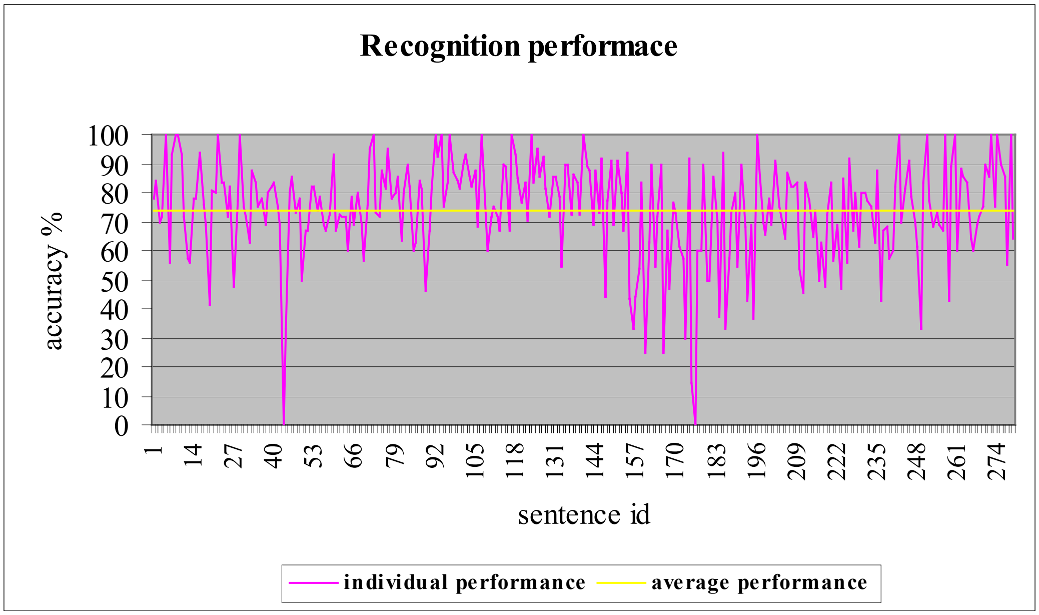


Figure 5-9: Recognition performance using dynamic language model (Strategy #2: without WordNet)

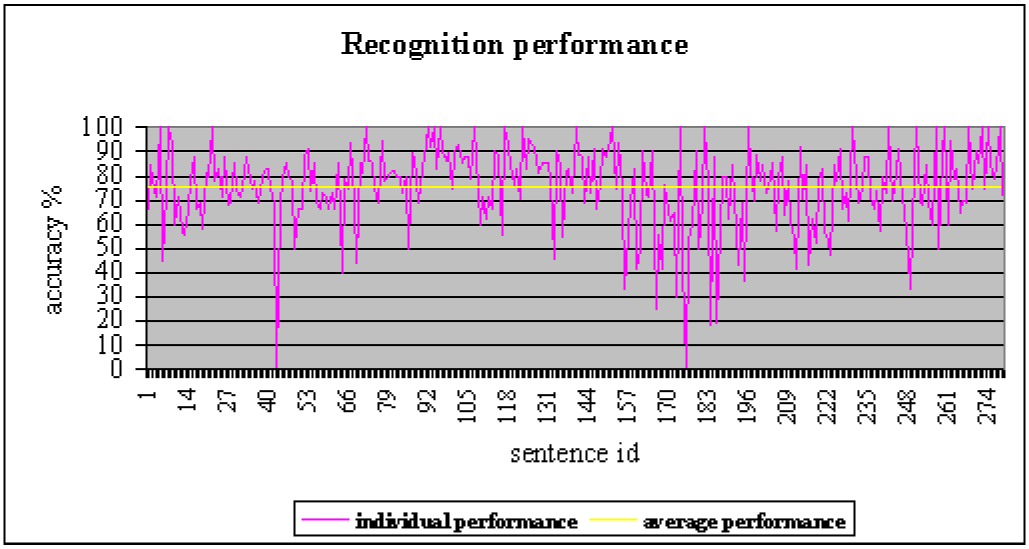


Figure 5-10: Recognition performance using dynamic language model (Strategy #2: with WordNet)

The experiments show the better performance (75.17%) of dynamic language model when we apply our second strategy. The dynamic language model using semantic clustering technique fails to improve the recognition performance in spite of huge computational cost.

The average performance comparison is shown in Fig. 5.11.

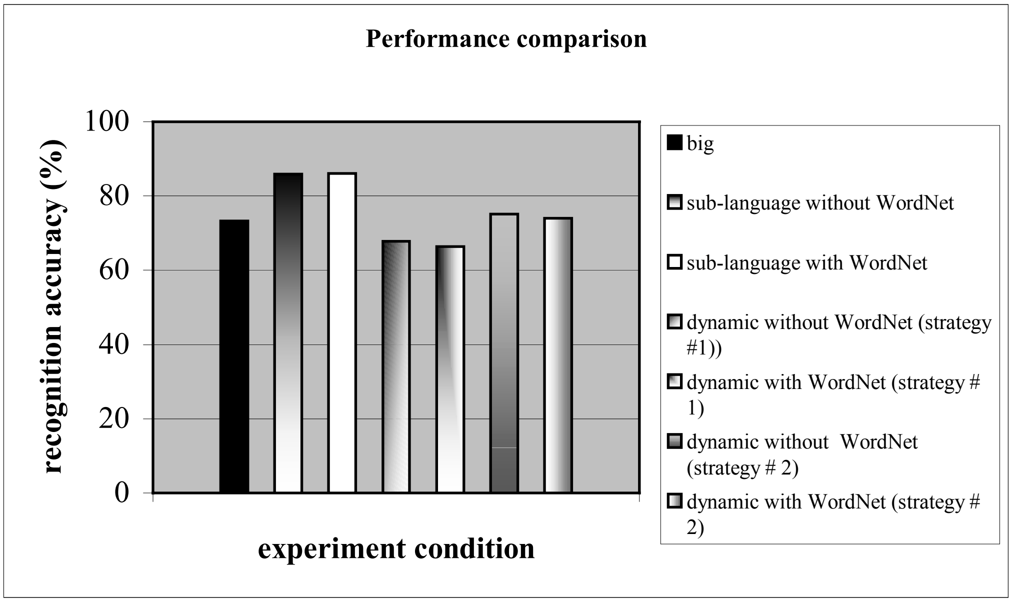


Figure 5-11: Performance comparison

Chapter 6
Conclusion and Future Work

Large vocabulary speech recognition system experiences performance degradation due to its vocabulary size. As the vocabulary increases, search path and the mutual confusion among the pronunciations of the words increase and consequently performance of the recognizer usually falls below the expectation level.

To improve the performance of ASR and to be able to have free and unlimited dictation systems, we used clustering technique, specifically self-organized map in generating topical language models. Since our speech recognition task is sentence based, we employed sentence level clustering. In the clustering process, we introduced a new idea in representing feature vectors. Conventionally, text is considered as a bag of words without any contextual relation and sentence is represented as a sequence of 1 or 0 or the word's frequency. In our proposed approach, sentences are encoded as a sequence of normalized word ids to preserve the long-term dependency among the words.

The rationale behind incorporating SOM is to distribute the whole corpus into a specific number of clusters according to the textual similarities. Since each of the clusters can be considered as a huge collection of similar texts, building language models out of the texts are expected to reduce the effect of data sparseness and language models are expected to be more efficient in terms of the recognition performances.

In order to verify the efficiency of the topical language models, a set of experiments has been performed, and we achieved consistently better recognition performance, compared with the performance obtained from the big language model. The results show gradual improvement in recognition performance with the increased number of clusters, but once the cluster size exceeds 20, the performance does not change significantly, rather it starts degrading. Since, there is always a trade-off between the cluster size and performance, cluster size as 20 has been chosen for whole experiments.

In order to enhance the performance of language model, we have integrated WordNet as a background knowledge source into the clustering process. From WordNet senses of noun, verb and adjective have been extracted. Noun has altogether 106,047 senses, deep into the 10 levels starting from parent. Verb has altogether 13,092 senses, deep into 5 levels, and adjective has 13,923 senses all are in the same level. Because of huge number of senses, we have applied a strategy to combine all of the senses (noun and adjective) down to the level 4 into one single sense. Since the assignment of words to senses in WordNet is ambiguous, and there is no matured solution yet available, the most frequent used senses are used in our research.

The efficiency of the topical language models incorporating WordNet has been verified for the cluster size 20. But unfortunately, no such significant improvement has been achieved in spite of huge computational cost. The reason behind this might be the disambiguation issue.

Although, topical language models without WordNet involvement shows a significant improvement in recognition performance ($\approx 86\%$), problem of automatic identification of suitable cluster(s), in other word suitable model(s) still remains as an active research area. In order to tackle the problem, first we tried to find a suitable model based on the perplexity measure. And secondly, we tried to generate a dynamic language model through combining closest cluster(s). But, both of the efforts provided very unsatisfactory performance. This leads to conclusion that perplexity measure is not reliable for a text with single sentence.

Finally, we tried to generate a dynamic language model through collecting all of the 20 responses, generated from the 20 models for a given utterance. Since, the tri-gram language model is built based on the probability estimation, the model assigns higher probability value for all the words frequently appear in the training data set. From the experiment, it has been found that words like, "*a*", and "*the*" (as well as other short stop words) appeared more frequently than other candidates. The results show a significantly poor recognition performance for the dynamic language models, built using CMU language model toolkit (performance $\approx 10\%$). Even using *quickLM* algorithm, recognition performance cannot reach up to 20%. But for our *modified quickLM* algorithm, performance reaches up to 67.8%.

So instead of using the sentences as string of words, we hyphenated all of the words in a sentence and made a sentence as a single connected word. In this way, we have 20 connected words for building the dynamic model. Since we have only 20 words in the dictionary, one of the words must appear when we pass the same utterance once again.

Based on this concept, we have performed experiment and found better results (75.17% for the best case). Although the results are far below the results obtained from the average performance (86%) of 20 clusters, it is better than the performance obtained from the big language model (73.3%).

In this thesis, we presented a way to extract the task-dependant acoustic model parameters. The method of extraction is a simply text processing that enables the system to consume low memory. This method is beneficial for small vocabulary system, especially for embedded system.

The work presented in this thesis addresses a number of basic issues. These are some of the ideas that can be pursued to improve the recognition performance.

- To investigate and implement various word-sense disambiguation strategies in the clustering process.
- To apply natural language processing technique and fuzzy inference in the speech recognition system.

Bibliography

1. Rabiner, L.R, Juang, B.H. “Fundamentals of speech recognition”, Prentice-Hall, 1993.
2. http://www.ling.mq.edu.au/units/slp806/unit_notes/langmodel.html - AEN790.
3. Samouelian et al. “1994 Knowledge based approach to English consonant recognition”, *Proc. Int. Conf. On Acoust. Speech & Signal Process*, pp 77–80, Piscataway, NJ.
4. X. D. Huang, Y. Ariki, and M. A. Jack. “Hidden Markov Models for Speech Recognition", Edinburgh University Press, 1990.
5. J. Picone. “Continuous speech recognition using hidden Markov models”. *IEEE ASSP Magazine*, pp. 26-41, July 1990.
6. L. R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”, *Proc. IEEE*, 77(2): 257-286, 1989.
7. D. Ray Reddy. “Speech Recognition by Machine: a Review”, *Proc IEEE*, 64(4): 501-531, 1976.
8. L. R. Rabiner and B. H. Juang. “An introduction to hidden Markov models”, *IEEE ASSP Magazine*, pp. 4-17, January 1986.
9. Claudio Becchetti and Lucio Prina Ricotti. “Speech recognition theory and C++ implementation”, John Wiley and Sons, 1999.
10. Frederick Jelnek. “Statistical Methods for speech Recognition”, The MIT Press, 1999.
11. Xuedong Huang, Alex Acero, Hsiao Wen Hon. “Spoken Language Processing”, Prentice Hall, PTR, 2001.
12. H. Bourlard and N. Morgan. “Connectionist Speech Recognition-A Hybrid Approach”, Kluwer Academic.
13. http://www.dcs.shef.ac.uk/~stu/com326/
14. R. P. Lippmann. “Review of Neural Networks for Speech Recognition”, *Neural Computation*, 1:1-38, 1989.

15. Aravind Ganapathiraju. “Suport Vector machine for speech recognition”, http://www.isip.msstate.edu/publications/books/msstate_theses/2002/support_vectors/thesis/thesis_final.pdf

16. Joseph W. Picone. “Signal modeling techniques in speech recognition”, *Proceeding of the IEEE*, Vol. 81: No. 9, September, 1993.

17. Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. “Class-based n-gram models of natural language”, *Computational Linguistics* 18:467-479, 1992.

18. Radu Florian, and David Yarowsky. “Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation”, *37th Annual Meeting of the Association for Computational Linguistics*, 1999.

19. WordNet: http://www.cogsci.princeton.edu/~wn/

20. Andreas Hotho Hotho, Steffen Staab, and Gerd Stumme. “WordNet improves Text Document Clustering”, Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany.

21. CMU Speech: http://www.speech.cs.cmu.edu/

22. Good, I.J. “The population frequencies of species and the estimation of population parameters”, *Biometrika*, vol. 40, pp. 237 - 264, 1953.

23. Katz, S. “Estimation of probabilities from sparse data for the language model component of a speech recognizer”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400-401, March 1987.

24. Ney, H; Essen, U; Kneser, R. “On the estimation of “small” probabilities by leaving-one-out”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, December 1995.

25. Kohonen, T. “Self-Organizing Maps”, Springer, Berlin, Heidelberg, 1995.

26. Haykin, S. “Neural Networks: A comprehensive Foundation”, Macmillan College publishing company, NewYork, 1994.

27. http://www.fact-index.com/h/ha/hamming_distance.html

28. Chirag Shah, Bhoopesh Chowdhary, Pushpak Bhattacharyya. “Constructing Better Document Vectors Universal Networking Language (UNL)”, *Proceedings of International Conference on Knowledge-Based Computer Systems (KBCS)* ,2002.

29. Bhoopesh Chowdhary, Pushpak Bhattacharyya. “Text clustering using semantics”, Indian Institute of Technology, http://www2002.org/CDROM/poster/79.pdf

30. Y.Yang, and J.P Pederson. “Feature selection in Statistical Learning of Text”, Report CMU-CS-97-127, 1997.

31. Thorsten Joachims. “A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization”, *Proceedings of the Fourteenth International Conference on Machine Learning*, p.143-151, July 08-12, 1997.

32. WebSOM: http://websom.hut.fi/websom/

33. Shiping Huang, Matthew O. Ward, Elke A. Rundensteiner. “Exploration of Dimensionality Reduction for Text Visualization”, 2003. http://citeseer.nj.nec.com/huang03exploration.html

34. Li Xiaobin, Stan Matwin & Stan Szpakowicz (1995). “A WordNet-based Algorithm for Word Sense Disambiguation”, *Proceedings of IJCAI-95*, Montréal, Canada.

35. http://sensei.lsi.uned.es/NLP/papers/NLDB01-wsd.pdf

36. Yarowsky, David. “Word sense disambiguation using statistical models of Roget's categories trained on large corpora”, *Proceedings of the 14th International Conference on Computational Linguistics*, COLING'92, 23-28 August, 1992, Nantes, France, 454-460.

37. HTK: http://htk.eng.cam.ac.uk/

38. quickLM: http://www.speech.cs.cmu.edu/tools/lm.html

39. Structure of Sphinx: http://www-2.cs.cmu.edu/~rsingh/sphinxman/scriptman1.html

40. BNC: http://www.hcu.ox.ac.uk/BNC/

41. CMU dictionary : http://www.speech.cs.cmu.edu/cgi-bin/cmudict

42. VOA News : http://www.voanews.com/specialenglish/

43. Utltralingua English Dictionary: http://www.ultralingua.net/