LARGE SPAN CONSTRAINT MODELS FOR SPEECH SYSTEMS

A THESIS

submitted by

A. NAYEEMULLA KHAN

for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JULY 2005

O my Lord! advance me in knowledge

Quran 20:114

Glory to Thee: of knowledge we have none, save what Thou hast taught us: in truth it is Thou who art perfect in knowledge and wisdom

Quran 2:32

To my wife A. Shahina

for her unflinching support.

To our parents

A. Abdul Shukur and A. Shahanaz

for their unlimited prayers and

immeasurable support.

THESIS CERTIFICATE

This is to certify that the thesis entitled **LARGE SPAN CONSTRAINT MODELS FOR SPEECH SYSTEMS** submitted by **A. Nayeemulla Khan** to the Indian Institute of Technology Madras for the award of the degree of Doctor of Philosophy is a bonafide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. B. Yegnanarayana

Dept. of Computer Science and Engg.

Chennai 600 036

Date:

ACKNOWLEDGEMENTS

I thank the Almighty for having blessed me with an opportunity to study in this esteemed Institution. I am ever grateful to my guide Prof. B. Yegnanarayana for taking me under his wings. It is he who made me reach my goals. He was the guiding figure, like the father who constantly helps a child learn to walk. He helped me back on the track in my research when I fell off the tracks. His dedication to research and discipline at work is a source of awe and inspiration for me to follow. I will always be thankful to him for his moral support and encouragement and for the valuable time he spent for me.

Dr. C. Chandrasekar has been helpful on various occasions. He encouraged me to be systematic and hard working. His critical questions on the various research issues were most helpful. I thank my doctoral committee member Dr. Hema A. Murthy for her help throughout my course of study. I benefited from her courses and the discussions we had.

I am very thankful to Dr. S. Rajendran who helped me in many aspects of research. I used his expertise in linguistics and Perl programming to my fullest advantage on many occasions.

I thank Prof. C. Pandurangan and Prof. S. Raman, successive chair persons of the department of Computer Science and Engineering, for their constant support, for providing me an opportunity to work in the department and for the excellent facilities made available for my research. My sincere thanks are due to my doctoral committee members, Dr. Deepak Khemani, Dr. Hema A. Murthy, Dr. Sukhendu Das, Prof. Sreesh Choudry and Prof. K. Radakrishna Rao for their valuable suggestion and time spent on evaluating my progress in research. I thank the non-teaching staff at the department who were always willing to help me in all administrative matters.

I am grateful to B. S. Manoj and T. Nagarajan for the numerous discussions and help I received during my initial course of study. I am also thankful to Satya Sai Prakash, Chitra Babu, Madhu and Sakthi Balan for the useful discussions we had.

My lab mates during this research period ensured that the work was not dreary and my stay pleasurable. I thank them all in no particular order. Suryakanth has given me constant moral and technical support. His eagerness to help everybody is amazing. Two people, in addition to others, who more often tried to decipher my writings are Guru and Dhananjay. Satish, Sriram and Chaitanya were just a call away, always ready to help. I had constant banter and chatter with all my colleagues KSR, Prasanna, Krishna Mohan, Kumarasamy, Venkatesh, Anil, Vinod, Palanivel, Suresh, Anand, Gupta, Ramesh, Sharat, Hegde, Kamakshi Prasad, Devaraj and Leena. I am thankful to each and every one of them for all their help.

I am indebted to my wife A. Shahina for her support, encouragement, love and for boosting my confidence whenever it was low. I have two fountains of joy, Almighty willing, that can never go dry. My children N. Safiyyah and N. Muhammad Aqeel are a source of immense pleasure. They light up every moment of our lives. They dispel the gloom and lift my spirits. Our parents A. Abdul Shukur and A. Shahanaz have been the constant source of prayer, support, motivation, succor and encouragement, propelling and driving me from the start to the completion of this work. Their support is like a backbone, without it I would collapse. I cannot thank them enough, so I leave it at it. All praises be to Allah for having blessed me with such a wonderful family.

A. Nayeemulla Khan

ABSTRACT

Keywords: *Speech recognition; syllables; language modelling; latent semantic analysis; large span constraint models, speech-based language model, speaker recognition; idiolectic characteristics;*

In spoken communication among humans, the message content of the speech is spread over a large duration of the discourse. The semantics of the message cannot normally be obtained by processing short segments of speech. For example, consider the sentence *"the doctor cut open the stomach during surgery"*. In this sentence, the words *doctor* and *surgery* are semantically related to each other in this context. This relationship exists even though the words are separated by a large number of words and cannot be captured by small windows of speech. Such constraints are called *large span constraints*. These large span constraints pertain to a language, and can be captured using language models in the framework of Latent Semantic Analysis (LSA). The LSA uses the information about the co-occurrence of words in text documents (set of semantically coherent sentences) to derive the large span constraints. Language models that use these large span constraints also perform better for most speech tasks. These models are called large span constraint language models.

Large span constraints exist in the speech signal as well. The semantic constraints among the words can be inferred from the language associated with the speech without the explicit transcription of the speech into text. Besides the semantic constraints, the speech signal contains information related to the associated language such as syntax, prosody, legal sound units and coarticulation constraints all of which are not available when a text corpus is used for language modelling.

Language models are conventionally developed from large text corpora. For many

spoken languages such corpora do not exist, especially for non-literary languages. For such languages, it is easy to record speech data in large volumes with minimal effort. The main issue addressed in this thesis is, whether a language model can be derived directly from the speech data. We show that it is indeed possible to do so using the framework of LSA. Due to the absence of a complete and reliable front end of a speech recognition system, the performance of the proposed models of large span language constraints is tested using perplexity measure. Perplexity is the measure of performance of a language model. The perplexity can be considered as the average number of words (units) that can follow a given word (unit) in a language. The smaller the perplexity of a model, the better is the ability of the model to predict the word sequence. When the large span constraint language model is combined with the standard bigram language model, it performs better than the bigram model alone. This model is similar to the combined bigram + LSA (bi-LSA) model normally derived from text transcripts.

The task of developing a speech recognizer for Indian languages requires a good recognition rate at the level of the subword units in speech. Using syllable as the subword unit, the usefulness of the large span constraint models at the syllable level is examined using both the speech and text corpus. We show that latent constraints exist even among the syllables as captured by the LSA model. The performance of the syllable level bi-LSA model is better than a syllable-level bigram model. The bi-LSA model can be used in a speech recognizer to improve its performance. The issue of errors in speech recognition need to be addressed while developing large span constraint models from speech. We study the effect of these errors by simulating the errors at the word level while developing the large span constraint model.

When a person speaks, his idiolectic characteristics are embedded in the speech.

The indicator words characterising the idiolectic traits of the speaker are often repeated and spread over the entire conversation. We propose an approach to capture the idiolectic characteristics of a speaker using large span constraint models.

To understand the issues in the development of a speech recognizer, the statistical characteristics of the basic subword units of a language are studied. A syllable-level recognizer is developed for two Indian languages. Some of the issues in modelling the dynamics of the syllables in the framework of hidden Markov models are discussed, and some approaches to improve the performance of the syllable recognizer are presented.

The main contributions of this thesis are summarised as follows:

1. Large span constraint language models are derived directly from the speech signal for the first time.
2. It is shown to a limited extent that large span constraints exist at the syllable level also. An approach is proposed to construct a large span constraint model at the syllable level.
3. The performance of the speech-based bi-LSA model is shown to be better than the bigram language models both at the word and syllable level. The performance is equivalent to and perhaps marginally better than the text-based bi-LSA model at the word level.
4. A large span constraint language model is proposed to capture the latent idiolectic characteristics of a speaker. It extends the use of the LSA concept for the speaker recognition task.
5. A support vector machine based preclassifier is proposed to reduce the search space in the speech recognizer and to improve its performance.
6. From an information theoretic perspective, it is shown that the syllable is an

appropriate subword unit for speech recognition.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AANN	–	Autoassociative Neural Network
ANN	–	Artificial Neural Network
ATIS	–	Air Traffic Information System
bi-LSA	–	Combined bigram + LSA language model
CART	–	Classification and Regression Tree
CVC	–	Consonant-Vowel-Consonant
CV	–	Consonant-Vowel
DCF	–	Detection Cost Function
DET	–	Detection Error Tradeoff
DP	–	Dynamic Programming
DTW	–	Dynamic Time Warping
EER	–	Equal Error Rate
GMM	–	Gaussian Mixture Model
HMM	–	Hidden Markov Models
IITM	–	Indian Institute of Technology Madras
ITRANS	–	Indian language TRANSliteration code
LM	–	Language Model
LPC	–	Linear Prediction Coefficients
LSA	–	Latent Semantic Analysis
LSI	–	Latent Semantic Indexing
LVCSR	–	Large Vocabulary Continuous Speech Recognition
MDS	–	Macro Demi-Syllable
MFCC	–	Mel Frequency Cepstral Coefficients
MLP	–	MultiLayer Perceptron

NIST	–	National Institute of Standards and Technology
NP	–	Noun Phrase
OOV	–	Out of Vocabulary
PLP	–	Perceptual Linear Prediction
POS	–	Parts of Speech
SOM	–	Self Organising Maps
SRI	–	Stanford Research Institute
SVD	–	Singular Value Decomposition
SVM	–	Support Vector Machines
TIMIT	–	Texas Instruments and Massachusetts Institute of Technology
VCV	–	Vowel-Consonant-Vowel

CHAPTER 1

LARGE SPAN CONSTRAINTS: AN INTRODUCTION

"The doctor cut open the stomach with a scalpel during surgery"

In this sentence it is observed that there exists a relationship between the words *doctor* and *surgery*. Such relationships arise due to the semantic constraints that exist among the words used in describing a particular concept or story. These constraints exist though the words are separated by a large number of words (8 in this case). Such constraints are called *large span constraints*. These large span constraints can be captured from the topic of discourse to express the semantic context without regard to the syntax of the words used. An approach to model these large span constraints is to use Latent Semantic Analysis (LSA). The LSA uses the co-occurrence information of words in a large text corpus to model the semantic constraints.

1.1 IMPORTANCE OF LARGE SPAN CONSTRAINTS

When humans want to communicate their thoughts among themselves either in spoken or written form, they make use of a language. Constraints exist at various levels among the different components of speech and language. The constraints in speech could be due to physiological characteristics of the speech production process, or the psychological characteristics that motivates the thought process behind the act of communication. The constraints could also be related to the characteristics of the language at the phonological, morphological, syntactic, semantic or pragmatic levels.

Every constraint imposed by the language and speech can be used to disambiguate among the many possible alternative hypotheses at each stage of decoding the message being conveyed. It is desirable to incorporate as many constraints as possible into the speech systems for better performance.

Large span constraints exist in speech and the associated language. They are implicit (i.e., latent) rather than explicit (i.e., governed by rules). These constraints are useful in language modelling, speech recognition, speaker recognition and many other speech systems. In a language, the large span constraints exist at different levels. For example, at the syntactic level, the idea of tagging the words in a sentence with Parts of Speech (POS) tags, is to group words into similar categories. This information is used to determine what words are likely in the vicinity of a word. It can also be used in a text-to-speech system to determine the correct pronunciation of a word. The large span constraints in terms of the POS grouping can be seen in the sample sentence considered earlier. It reveals that the words *doctor* and *scalpel* would come under the category of Noun Phrase (NP). Likewise, a parsing of much larger sentences also would reveal the underlying large span relationships that exist among the words and phrases. At the semantic level, as we saw earlier, the words *doctor* and *surgery* are semantically related, though the words are separated by a large span. If the sentence is followed by *"whereas the scheduled surgical procedure was an open heart surgery"*, we now understand from this discourse that the original surgical procedure was a mistake. This correct interpretation of the discourse can be arrived at only when all the words and the large span constraints/relationships among the sentences and words are taken into account. Thus we see that large span constraints exist at various levels in a language.

The large span constraints also exist in speech. If the previous example sentence is

spoken, the listener still understands that the words *doctor* and *surgery* are semantically related to the discourse, though the words are far apart and no textual representation is used. Other large span constraints like intonation in speech are useful in decoding the message content in the speech signal. Likewise, when a person speaks, idiolectic traits are embedded in the conversational speech. Frequent use of back channels like *umm...*, *lip smack* or phrases like *"I mean"* are idiolectic traits specific to a speaker, and they manifest over large spans in conversational speech. These large span constraints are useful in speaker recognition studies. In applications involving dialogue modelling, the large span constraints play a vital role in disambiguating between different possible interpretations of the user response. Thus we see that large span constraints exist in speech and language, and these constraints are useful in many speech systems.

1.2 CONSTRAINTS IN A LANGUAGE

The language associated with speech can be analysed at various levels. The purpose of analysis is to derive constraints that can be used in various speech systems. (a) At the phonetic level, the characteristics of the basic sound units (say, phonemes) in a language, and the similarities among them is dealt. (b) At the morphological level, the way words are built from the basic meaning bearing stem of the word is analysed. Morphology together with phonetics is useful in developing a good lexicon for use in a speech recognizer. The lexicon is used to describe the words in terms of the component phones. Ambiguities arise out of the acoustic-phonetic analysis of the speech recognizer. This can often be resolved by subjecting the word hypotheses of the speech recognizer to syntactic, semantic and pragmatic analysis. These types of analysis incorporate higher levels of knowledge which provide additional constraints to disambiguate among the possible hypotheses of words in the recognizer. (c) Syntactic

knowledge is used to determine whether a particular sequence of words can occur in a grammatically correct sentence. (d) Semantic level constraints can be used to determine whether the syntactically correct sentence is actually meaningful. Semantic knowledge can also be used to predict words and phrases that are similar in meaning to the current context. (e) Pragmatic constraints are used to disambiguate if the meaningful sentence is appropriate in the context of the ongoing dialogue. The syntactic, semantic and pragmatic constraints can extend over large spans, and can be called as large span constraints.

1.3 CONSTRAINTS IN SPEECH

Speech contains constraints at various levels. When a person listens to speech in a language, he uses all the knowledge (syntactic, semantic and pragmatic) of the language that he has acquired to decode the message being conveyed. If the listener hears speech in an unknown language, since he/she does not know the legal sound units of the language spoken and neither the syntactic, semantic or pragmatic constraints of the language he/she is unable to decode the message. Each of these knowledge sources can be viewed as constraints useful to decode the message. These large span constraints are essential for decoding speech. Speech implicitly contains the identity, gender, emotional state and stress level of the speaker, and the language spoken, in addition to the message being conveyed. Each of these information sources can also be considered as constraints derivable from speech. Many of these constraints are not present in the written (text) form of the message. In any speech system it is desirable to incorporate as many of these constraints as possible to improve its performance.

Speech can be analysed (based on the frame size) at the sub-segmental (<5 msec), segmental (10 to 30 msec) and suprasegmental (>100 msec) levels. These divisions are

in relation to the normal values of the fundamental frequency or pitch period (5 to 10 msec). The purpose of analysis at the different levels is to understand the constraints that exist at these levels, so that they can be used appropriately in various speech systems. At the suprasegmental level, the variation in intonation, duration of the syllables or words, the stress on a syllable, the speaking rate of the syllables and the influence of one sound unit on another (coarticulation effects) can be studied. The suprasegmental features display large span relationships. The suprasegmental or prosodic features also carry constraints related to phonology, morphology, syntax, semantics and pragmatics. At the segmental level the spectral characteristics in short segments of speech are captured by various types of features like Linear Prediction Coefficients (LPC), Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction coefficients (PLP). These features predominantly capture the local constraints. If wider windows are used in PLP type of features like RASTA-PLP, large span constraints are indirectly incorporated. The segmental features reflect the changes due to phonetic contexts. The subsegmental analysis is mainly guided by the speech production mechanism. It aims at bringing out the local constraints. When pitch synchronous analysis is performed at the subsegmental level, it may be possible to capture the consistent variations in similar segments of successive glottal cycles. It also helps to reduce the effect of the fundamental frequency on the linear prediction analysis. Constraints at the suprasegmental, segmental and subsegmental level are not completely independent of each other. For example, constraints at the suprasegmental level like intonation patterns or stress on syllables are known to influence the characteristics of the voice source in continuous speech. These constraints provide important cues at different levels which can appropriately be used in various speech systems.

1.4 MODELLING CONSTRAINTS FOR SPEECH SYSTEMS

1.4.1 Modelling large span constraints in a language

When large span or local constraints are used to model a language, then such models are also called language models. Language models conventionally have always been derived from a text corpus. Statistical language models are widely used in most speech and natural language applications like speech recognition, machine translation, optical character recognition and information retrieval. The most widely used statistical language model is the n-gram model. The n-gram models use the history of the previous $n-1$ words to predict the current word. For n-gram models, if more history (large n) is known, the better is the prediction. The difficulty in using models with large n is the need for large amounts of data. For example, using a moderate vocabulary of 2000 words and a trigram model ($n = 3$), the number of possible word combinations is about 2000^3 or 8 billion. The probabilities of all these large number of parameters needs to be estimated from a finite text corpus. Though all such combinations are not permissible in a language, the number of possible parameters in the model is still large. It has been observed that even with the best smoothing techniques and backing off of the language models, the performance of n-gram models peak for $n = 5$ [1]. Informal estimates by IBM suggest that bigram models peak at 250 million words [2]. The n-gram models primarily capture the local constraints. In the example sentence considered earlier, to bring out the dependency between *doctor* and *surgery* using an n-gram model, one would require $n = 11$. This is clearly not feasible using the current corpus sizes and estimation techniques. But large span constraints in speech and language can be effective in disambiguating/predicting the hypotheses output by a speech recognizer and need to be modelled.

An approach to modelling large span constraints in a language is using latent semantic analysis. The LSA is widely used in information retrieval studies [3] [4], and in models of human cognition and learning [5]. The LSA technique uses the knowledge of the co-occurrence of words in text passages/documents to infer the possible semantic relatedness of the words in a document. An LSA model derived from the text corpus is shown to improve the performance of the language model in terms of reduction in perplexity [6] [7]. When these large span constraint models are used in tandem with a speech recognizer, the error rates of the recognizer are reported to have reduced [8].

One basic requirement to develop a language model is the availability of a large text corpus. Language models are generally domain dependent. For example, a language model developed using the corpus for news transcription task cannot be used for a medical transcription task. As the domain is different, the vocabulary and style of discourse would be different, causing the language model to perform poorly. Also, it is generally difficult to obtain a large clean text corpus of medical transcription. In all similar tasks it is likely that the audio data pertaining to the domain under consideration (say medical transcription) running to thousands of hours may be available. The situation is also similar for dialogue modelling applications, where the text corpus has to be derived manually by listening to the dialogue or is to be obtained from a preliminary speech recognizer. In India there are 3372 languages and dialects spoken (as per census 1991) [9]. Many of these languages are non-literary. For non-literary languages a text corpus is not likely to be available for developing language models. In all such situations, and even in the case of literary languages, it is much easier to record speech data for any desired volume. Given such a database, it is desirable to derive a language model capturing local and global constraints directly from the speech signal. Investigating this idea constitutes a major contribution of this work.

1.4.2 Modelling segmental constraints in speech

In speech recognition an n-gram language model or a large span constraint language model is used to constrain the recognizer and recover from errors in recognition. The underlying principle is to use higher level linguistic knowledge to recover from the errors. The basic requirement for this approach to work is a good recognition of the basic sound units in the speech utterance. Also, the word hypothesis of the speech recognizer should be 'fairly good'. Otherwise no amount of linguistic or domain specific knowledge can help to recover from the basic flaws in the recognition hypothesis.

Study of the characteristics of the sound units in a language is the first step in designing a speech recognizer. The statistical properties of the sound units is studied in Chapter 3. There are different types of sound units like phonemes, biphones, triphones, demisyllables, syllables, words, etc., that can be used in a speech recognizer. Each type of unit captures constraints to varying degrees. The biphones and triphones are artificial units created from a machine recognition perspective. They are designed to incorporate local contextual constraints in them. The syllables are naturally occurring linguistic units. A unit that is linguistic in nature and inherently capable of capturing the constraints at the level of sound units is desirable. An information theoretic approach can be adopted to determine the most suitable subword unit for speech recognition. This study shows that syllable seems to be an appropriate unit for speech recognition, as they capture the constraints better.

Acoustic modelling of the syllables to capture the segmental constraints can be done using different techniques like Support Vector Machines (SVM), Hidden Markov Models (HMM) or Artificial Neural Networks (ANN). Each modelling technique attempts to capture the variability in the speech sounds. Syllables are large dynamic units. The hidden Markov model can be appropriately used to capture the dynamics

of the syllables. Once a syllable recognizer is developed with good recognition performance, then a word level or sentence level recognizer can be developed. In this work, we focus on the development of a syllable recognizer.

1.5 SCOPE OF THE THESIS

One of the main objectives of this thesis is to derive a large span constraint model directly from the speech signal, and examine the performance of this model in relation to the large span constraint model derived from a text corpus. Methods are proposed to derive large span constraint models from the speech signal. To understand the effect of errors in speech recognition on the performance of language models, the performance of the large span constraint model is examined for erroneous transcriptions (text) at the word level. An efficient syllable recognizer is also developed for two Indian languages in the framework of HMM. Using a transcribed text corpus for the Indian language Tamil, large span constraint models are also examined at the word and syllable level. For the syllable level model the span length is limited to a small number of syllables.

Studies on modelling the large span constraints are extended to capture speaker-specific information from a corpus of large but somewhat erroneous transcription of speech. The evidences from these large span models are combined with the evidence obtained from other speaker recognition systems using speech-based features to improve the speaker recognition performance significantly.

1.6 ORGANISATION OF THE THESIS

The evolution of ideas presented in this thesis is given in Table 1.1.

Chapter 2 deals with the review of related work in developing large span constraint models from a text corpus, and the use of LSA concept in information retrieval studies.

The chapter also reviews work related to recognition of syllables, and the use of large span constraints for the task of speaker recognition.

In Chapter 3 the development of the speech database for Indian languages is described. The database is analysed to study the statistical characteristics of the sound units in the languages from a speech recognition perspective. The issue of choice of subword unit for speech recognition is also addressed in the chapter. The syllable seems to be an appropriate unit for speech recognition from an information theoretic perspective. The development of a syllable recognizer for Indian languages is discussed in Chapter 4. The approaches to improve the performance of the basic syllable recognizer are also discussed in the chapter.

Large span constraint models are developed using the LSA framework for a test corpus, both at the word and syllable level in Chapter 5. The effect of erroneous transcripts (as expected in transcribed speech) on the performance of these models is discussed in Chapter 6.

Chapter 7 deals with the development of speech-based large span constraint models at the word and syllable levels. Approaches to model the latent idiolectic speaker characteristics using the large span constraint models is described in Chapter 8. Chapter 9 summarises the work presented in the thesis, and lists some directions for future work.

Table 1.1 Evolution of ideas presented in the thesis.

Large Span Constraint Models for Speech Systems

- Development of a speech recognizer for Indian languages
  - Choice of subword unit: syllable, based on information theoretic criteria
  - Modelling the dynamics of the syllable, improving the recognition rate
- Performance improvements to the syllable recognizer
  - Reduction in search space- Neural network-based preclassifiers
  - Incorporation of higher level linguistic constraints: n-gram models, large span constraint language models
- Large span constraint language models from text corpus
  - Semantic constraints: LSA-based models for incorporating large span semantic constraint
  - Large span constraint models
    - Word level
    - Syllable level
- Large span constraint models from erroneous transcripts
  - Effect of errors due to speech recognition on the large span constraint models
- Speech-based large span constraint models
  - DTW-based approach
  - Template-based approach
- Extension of large span constraint models for other speech systems
  - Large span constraint models for speaker recognition

CHAPTER 2

REVIEW OF APPROACHES TO INCORPORATE CONSTRAINTS IN SPEECH SYSTEMS

2.1 INTRODUCTION

In this chapter, the previous works on modelling short span and large span constraints in speech and language for use in language modelling, speech recognition and speaker recognition systems are reviewed. The task of statistical speech recognition is to automatically transcribe the speech signal into a sequence of words W. Assume that the signal is a sequence of symbols $\mathbf{O} = o_1, o_2, \ldots, o_m$. The symbol o_i can be thought of as acoustic feature vectors generated in time as denoted by the index i. Let $W = w_1, w_2, \ldots, w_k$ denote a string of words, each belonging to some fixed vocabulary. If we denote $P(W|\mathbf{O})$ as the probability that the words W were spoken, given the observation sequence $\mathbf{O}$, the recognizer should try to find the word string W^* such that

$$W^* = \arg \max_W P(W|\mathbf{O}) \tag{2.1}$$

That is, the recognizer should find the most likely word sequence given the observed acoustic sequence. It is difficult to compute $P(W|\mathbf{O})$ directly, hence using Bayes formula, $P(W|\mathbf{O})$ can be written as

$$P(W|\mathbf{O}) = \frac{P(\mathbf{O}|W)P(W)}{P(\mathbf{O})}, \tag{2.2}$$

where $P(W)$ is the probability that the word string W is spoken, $P(\mathbf{O}|W)$ is the probability that when a speaker says W, the acoustic evidence is $\mathbf{O}$. Thus,

$$\begin{aligned} W^* &= \arg \max_W \quad \frac{P(\mathbf{O}|W)P(W)}{P(\mathbf{O})} \\ &= \arg \max_W \quad P(\mathbf{O}|W)P(W) \end{aligned} \tag{2.3}$$

The $P(\mathbf{O})$ can be ignored since we are maximising over all possible W and $P(\mathbf{O})$ is constant for a given input/test data. The likelihood $P(\mathbf{O}|W)$ is called the acoustic model, and the prior probability $P(W)$ is called the language model [10, 11]. The acoustic model $P(\mathbf{O}|W)$ can be estimated using hidden Markov models, hybrid ANN-HMM systems or similar systems [12]. Hidden Markov models are widely used in most current speech recognizers for acoustic modelling. These models have their limitations. But with appropriate design it is possible to incorporate, (1) the phonetic characteristics of the unit being modelled, (2) the duration of the sound units (2^{nd} order models), (3) a fixed duration for a state if desired, (4) contextual information (diphone models) and (5) discriminative training procedures. The language model is usually estimated by statistical language modelling techniques. It is used to constrain the speech recognizer. The constraints could be of short or large spans. The larger the number of constraints, the easier it is to recover from errors in the recognition process. It is desirable to incorporate the short and large span constraints into both the acoustic and language models.

In this chapter we review the work done in selecting an appropriate unit for speech recognition, with a focus on modelling the dynamics of the units in Section 2.2. The review of HMM-based approaches to speech recognition is given in Section 2.3. The techniques adopted for modelling large span constraints are discussed in Section 2.4. The development of LSA along with its use for language modelling is described in

Section 2.5. Work on the use of large span constraints in speaker recognition studies is discussed in Section 2.6.

2.2 CHOICE OF SUBWORD UNIT FOR SPEECH RECOGNITION

To recognize the words in a speech utterance, a model for each word in the vocabulary would be a simplistic and ideal choice. Word based systems provide good performance for task-specific applications having a small and fixed vocabulary [13]. For large vocabulary continuous speech recognition task, this approach is not feasible due to the lack of sufficient training examples to develop word level models that capture all the variability in speech. The alternative is to use subword models. As the subwords are smaller units compared to words, it is likely that there may be sufficient examples to train these models. The most popular choice of subwords used in the literature are the phonemes. Due to the significant coarticulation between phonemes, the recognition rate of phonemes has been poor. To model the local coarticulation and contextual constraints that manifest at the (segmental) level of sound units, larger (than phoneme) size subwords like syllables, triphones, diphones, and large-sized lexical or acoustically derived units have been tried. In this section we trace the previous works in the exploration of suitable basic units for speech recognition.

2.2.1 Units based on machine recognition perspective

A demisyllable contains the second half of the first phoneme and the first half of the next phoneme. The unit is designed to model the coarticulation effects. Using demisyllable as a base unit, a set of domain-specific larger units called macro-demisyllables (MDS) can be created [14]. These MDS units are formed by concatenating the demisyllables, and can grow up to the size of words or even phrases using an iterative

procedure. The size of the MDS is restricted such that they occur frequently enough to train HMM models for each unit. The MDS is good at capturing local coarticulation constraints [14]. When the size of the unit is large, it can even capture some amount of large span constraints among the demisyllables comprising the MDS unit. The performance of the MDS units which are trained in a context independent fashion is similar to a recognizer trained using context dependent triphones. For highly inflected languages, deriving a good pronunciation lexicon for use in a Large Vocabulary Continuous Speech Recogniser (LVCSR) is a difficult task, as there are many out of vocabulary words. Morphemes are the smallest meaningful units that can be derived morphologically. Morphemes as subword units provide a feasible solution, but the number of morphemes could be very large. In [15] small morphemes were merged using knowledge of the linguistic Parts Of Speech (POS) to provide more local contextual constraints. This improved the recognition rate of the morpheme units. The author also reported that if syllable is used as a subword unit instead of large morpheme-based units, the out of vocabulary rate was nearly zero. The recognition rate of the syllable-based unit however was poorer than morpheme-based units. Alternative subword units were also proposed in [16–18]. The different choices of subword units are driven from a machine recognition view point. They are tailored to suit the model being designed or to overcome the limitations of the existing modelling techniques.

2.2.2 Units based on acoustics

In spontaneous speech, which is highly coarticulated, it is difficult to map a linguistically fixed phoneme sequence to the acoustics. If a set of subword units can be derived from the acoustic signal and a pronunciation lexicon is constructed in terms of these acoustic segment units, it may form a viable alternative to phoneme-based systems. In

[19] an automatic method for word model generation based on acoustic segment units was proposed. It relied on a technique developed in [20] to derive the acoustic segment units. The recognition performance using this technique was reported to be better than a phoneme-based system for spontaneous speech. Alternatively, the acoustic unit derived could also be like multigrams as described in [21]. The multigram is a variable-length sequence of elementary acoustic observations. For each multigram, depending on its frequency of occurrence, a HMM model was constructed. Using the multigram HMM models, the sequence of multigrams in the signal was derived using the Viterbi algorithm. The multigram symbol sequence and the actual phonetic transcription of the signal were used to align the two transcriptions. The alignment was used to construct a lexicon based on the multigram sequence [22]. This approach was found to be better than a triphone-based recognizer, when the language model constraints were not used. The multigram sequence approach was also used for language modelling [23, 24]. The approaches mentioned above tried to derive subword units directly from speech, and incorporate more constraints at the subword unit level. It would be better if the unit modelled for speech recognition is motivated based on how humans produce and perceive speech. The syllable is one such unit.

2.2.3 Units based on speech production and perception mechanism

The syllable has a central vowel called the nucleus. The nucleus may be optionally prefixed or suffixed by one or more consonants, termed as the onset and coda, respectively. Syllable as a subword unit is intuitive for representation of speech sounds. Syllables are inherently natural units that are linguistically motivated. There is a close relation between the syllable and the speech production and perception mechanism [25–28]. Perceptually listeners are able to identify syllable boundaries in a word with a high

degree of interlabeler agreement [26]. Some of the advantages of using syllables as a basic unit for automatic speech recognition system are as follows [29].

1. The human auditory system integrates time spans of about 200 msec of speech [30], which corresponds roughly to the duration of syllables [26]. Thus the robust human perception can be modeled more accurately by the use of syllables instead of phonemes.
2. The relative duration of syllables is less dependent on variation in speaking rate than the relative duration of the phonemes [26].
3. The system will be robust against changes in speaking rate, speaking style (spontaneous vs continuous speech) and channel distortion (additive or multiplicative noise), especially if these changes are not seen during training [14].
4. Syllable boundaries are more precisely identified than phonemic segment boundaries both in the speech waveform and spectrographic display [31].
5. Syllables are inherently of longer duration and they capture the coarticulation between sounds better than phonemes.

A syllable-based recognizer was proposed for English in [32]. The approach showed an improvement over a monophone recognizer for a medium sized vocabulary. The authors also suggested that the number of training examples required for a syllable model may be higher due to the larger size of the unit. Towards obtaining an ideal large unit for Greek speech recognition a syllable based approach was adopted. A set of syllables with different vowel and phoneme combinations were used to obtain an improved performance over a comparable phoneme system [13]. In certain situations, it may be possible to combine the advantages of different modelling techniques to arrive at a better performance for the task at hand. In [29] it was shown that an

ANN-HMM hybrid system that used syllable as the subword unit, outperformed a phoneme-based system in classification of the subword unit in terms of frame error rate (frames misclassified by the ANN). On the *'OGI numbers'* task [33] for cross-database test, the syllable-based system showed robustness against changes in speaking style, and outperformed the phoneme-based system, indicating that the syllable can model the variability better. It is also possible to view the phoneme and syllable-based systems as two different approaches which can be combined for improving the overall performance. In [34], syllable and phoneme systems were combined at three different levels, namely, frame level, syllable level and utterance level. The study showed that a combination at the syllable level reduced the word error rates. This may be due to the fact that there exists some temporal organization of speech at the syllable level. An alternative design for the recognition system is to use a two-stage process. In the first stage, phoneme or syllable graphs are computed, which are domain-independent. The result of the first stage is then passed on to a domain-dependent second stage to determine the best word hypothesis. In such an approach proposed in [35], it was found that syllable representation with a bigram or trigram language model provided better constraints than a phonetic representation with higher n-gram language model (up to 6-gram). The performance of this approach was found to be similar to that of a conventional single-stage word-based recognizer.

Modelling the temporal dynamic behavior of the speech signal is complicated. If the information regarding transition regions is incorporated during modelling, it would lead to a better word recognition performance [36, 37]. Different approaches exist to model the transition regions, like (a) using diphone transitions [38]. (b) At the segment boundaries there is a rapid change in spectral characteristics of the signal. More features can be extracted around these segment boundaries by increasing the frame

rate to capture the change [39]. (c) The vowel onset point can also be used as an anchor to capture the dynamics of the syllable [40]. As a segment-based approach to model larger units of speech close to the size of syllables was proposed in [41]. Modelling these large syllable-like units captures the dynamics within phoneme boundaries and in the transition regions. The performance of the segment-based system was similar to that of a equivalent frame-based system.

Most Indian languages are syllabic in nature. Previous work on speech recognition for Indian languages had focussed on modelling syllable-like units. The focus had been to model the dynamics of the large size syllable-like units using approaches anchored around the vowel-onset point [42, 43]. Compression of feature vectors for efficient representation of the syllables had been proposed for use in ANN classifiers [44]. Different modelling techniques like ANN models [45], modular neural networks [28], constraint satisfaction neural networks [46], HMMs [28, 37, 47] and support vector machines [48] had been adopted for the task.

The suitability of modelling large subword units like syllables for the speech recognition task was discussed in this section. In the next section we look at some of the approaches to recognize syllable-like units.

2.3 ACOUSTIC MODELLING OF SUBWORD UNITS FOR CONTINUOUS SPEECH RECOGNITION

One of the basic problems in speech recognition is to match the acoustic patterns of two words. The matching technique used must account for the contextual phonetic effects like coarticulation and the large span constraints, like position of the sound in the whole sentence. It must also accommodate differences between speakers due to sex, age, speaking accent etc., intra-speaker variation due to stress, speaking rate and

environmental conditions. Techniques used for recognition vary from template matching, dynamic time warping, hidden Markov models, segment based models and neural network architectures to hybrid ANN-HMM approaches. Most state of the art LVCSR systems make use of hidden Markov models for acoustic modelling. Each of the above systems could be improved in a number of ways. Segment based models and neural network based connectionist approaches overcome some of the assumptions made in HMM-based modelling (discussed later in this section). There is no conclusive evidence that connectionist speech recognition algorithms are better than other approaches [49]. Except for the acoustic likelihood estimation, all systems use the framework of HMM to combine linguistic and acoustic information into a single network representing all possible sentences [12]. Comprehensive reviews on large vocabulary continuous speech recognition with insight into current trends and promising areas of improvement are provided in [12, 50]. A tutorial type introduction for connectionist approach to speech recognition is given in [49, 51]. A similar review on HMM-based speech recognition is given in [52, 53]. In this section we briefly review previous work in continuous speech recognition using HMMs.

Early speech recognition systems used the dynamic programming (DP) approach. In dynamic programming approach, time normalisation and pattern matching are accomplished in a single discrete optimisation procedure. In the DP approach the incoming speech signal was decomposed into a sequence of feature vectors and matched against precomputed reference vectors [52]. An unconstrained end-point DP algorithm proposed in [54] had a limited performance. Use of speech knowledge and imposition of local continuity constraints and global constraints on the time normalisation process improved the matching performance. This was one of the approaches where segment level constraints were directly used for automatic speech recognition.

Most current successful automatic speech recognition systems use hidden Markov models for acoustic modelling [55]. The purpose of an acoustic model is to calculate the likelihood of a feature vector sequence $\mathbf{O}$ given a word w. The HMM is a finite state machine in which at every time instant t that a state q_t is entered, an acoustic speech vector o_t is generated with an associated probability $b_{q_t}(o_t)$. The state vector q_t can take any one among the N possible states ($q_t = 1, 2, \cdots, N$). The probability of transition from state q_t to state q_{t+1} is given by $a_{q_t q_{t+1}}$. The joint probability of the vector sequence $\mathbf{O}$ and state sequence $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ given some model λ, is calculated as the product of the transition probabilities and output probabilities

$$P(\mathbf{O}, \mathbf{q}|\lambda) \;=\; \pi_{q_1} b_{q_1}(o_1) \prod_{t=1}^{T-1} a_{q_t q_{t+1}} b_{q_{t+1}}(o_{t+1}), \tag{2.4}$$

where q_1 is the entry and q_T the exit state, and π_{q_1} is the probability of the initial state. The model λ is specified in terms of $\pi = [\pi_1, \pi_2, \cdots, \pi_N]$, the state transitional probabilities $A = [a_{ij}]$, and the observation symbol probabilities $B = [b_j(o_t)]$. The observation sequence $\mathbf{O}$ is generally known, but the underlying state sequence q is hidden. Hence these models are called hidden Markov models [50]. The required probability $P(\mathbf{O}|\lambda)$ can be obtained by summing (2.4) over all possible *(NT)* state sequences using a forward-backward algorithm [11], where N is the number of states and T is the total number of symbols or feature vectors in the observation symbol sequence.

Some of the advantages of a HMM-based approach for speech recognition are [56]: (1) It provides a tractable mathematical framework that can be studied analytically. The model can easily generalise to unseen data. The HMM framework can take into account constraints imposed at the subword, syntactic and semantic levels. These constraints could be local or extending over large spans. (2) The statistical HMM approach imposes no particular structure, but provides sufficient degrees of freedom to

acquire the details during training. This is in contrast to knowledge-based approaches that attempted to build a general model of speech by listing every important aspect in detail, which were not too successful. However, specialised knowledge of speech has been used to constrain the models [57]. (3) The other important aspect is the flexibility of the model. It can be tailored such that the model topology and number of states match the typical duration, spectral complexity and variability of the sound being modelled [58].

There are certain assumptions made in the modelling of HMMs that are in variance with our knowledge of speech [56].

1. *Piece-wise stationarity assumption:* The HMM framework assumes that a speech pattern is produced by a piece-wise stationary process. But it is known that the speech patterns are derived from signals produced by a continuously moving physical system, the vocal tract system. The effect of this assumption can be reduced by modelling segments of speech. That is, by increasing the number of states, a sequence of piece-wise stationary segments may better approximate the dynamics [59].

2. *Independence assumption:* It is assumed that the probability that a given acoustic vector corresponds to a given state of the HMM depends only on the vector and the state, and is independent of the sequence of acoustic vectors preceding and succeeding the current vector and state. Thus the model takes no account of the dynamic constraints of the physical system which generated the sequence of acoustic data.

3. *State duration distribution assumption:* It is also implicitly assumed that the probability of a model staying in the same state for several frames is determined by the self loop transition probability. Thus the state duration in HMM

conforms to a geometric probability density function, which assigns maximum probability to state duration of one instant and smaller probabilities to longer durations.

The HMM-based systems have been built at the word level [11] [60], syllable level [61], and phoneme and triphone levels [62]. Syllables form ideal subword units for recognition as mentioned in Section 2.2.3. As the syllables are larger sized units the number of such syllables in a language (>5000) is also higher as compared to phonemes. But it is lesser than similar sized triphones in the language. Due to the large sized syllable inventory, more volume of data is required to get sufficient training examples for the rarely occurring syllables. Good performance at the subword level is essential for high overall recognition rate. In this thesis we focus on approaches for recognition of syllables using HMMs. An approach to recognizing syllables in continuous speech is to first identify the syllable boundaries (segmentation), and then perform recognition within the identified boundaries. Discrete HMMs and MultiLayer Perceptron (MLP) in combination with some heuristics can be used to spot vowels. A HMM-ANN system can then be used to identify the syllables between the vowel boundaries as discussed in [63]. The hybrid HMM-ANN systems have the discriminating power of ANN and also model the temporal variability. In this framework, using syllable as a unit, the recognition system outperforms a comparable phoneme-based system [29].

Presegmenting the speech signal into syllables prior to recognition was attempted as early as in 1975 [64]. Group delay based segmentation of speech into syllabic units followed by recognition of the syllables using HMM for Indian languages was proposed in [65]. The vowel onset point was also proposed as the basis for determining syllables in continuous speech [66]. This approach showed a marked improvement in recognition rate over a plain HMM-based system [67].

For the Greek language, the design of a syllable-like unit as a combination of multiple phonemes was reported to improve the performance by 9% over a phoneme-based system within the HMM framework [13]. A HMM-based syllable recognizer for the English language was shown to outperform a monophone recognizer by over 25% [32]. The stop-consonant vowels are some of the most difficult sound units to model due to variation in the acoustic signal. These stop sounds are studied extensively. Stops in Vowel-Consonant-Vowel (VCV) units were modelled as segments using continuous density HMMs. A high recognition rate (84.4%) was reported for a set of 5 stop sounds excised from continuous speech [68]. An approach to recognize of the stop consonants based on phonetic feature description of the speech signal using the HMM framework is reported in [69]. It assumed that the underlying Markov chain in the model was able to track the temporal evolution of the features [70]. The word initial and word final stop consonants were modelled at the subphonemic level (microsegmental level). This approach reduced the word error rates on CVC word list by 35%, when compared to a system using a single HMM for each stop consonant.

In languages like Japanese and Chinese, which are syllabic in nature, syllable has been widely used as the subword unit for modelling. For the Chinese tonal language, an approach to individually recognize the syllables and the tone was proposed in [71]. The recognition of the base syllables was modified periodically to take into account the tonal likelihood computed from a separate HMM of tones. The performance of the system was comparable to the standard approaches. It was achieved at a reduced computational cost. In a similar work, to simplify the recognition of syllables in LVCSR, a set of 416 base syllables were identified. These syllables, in combination with the tones, constituted the 1345 tonal syllables of the Chinese language [72]. A separate tone recognizer that recognized four different tones was developed. The 416

syllables were further decomposed into INITIAL consonant of a syllable and FINAL vowel with nasal or medial ending. A continuous HMM was used to model these decomposed units. A high recognition accuracy of 92.2% was reported for decoded Chinese characters. A similar subsyllable approach for Chinese was reported in [73]. A syllable-based LVCSR system for conversational telephone speech in English was reported in [74]. The system performed marginally better than a comparable triphone-based system for an evaluation on the switchboard corpus. A word error rate of 49.1% was reported. The syllable-based system outperformed the triphone system by nearly 20% on the alpha-digit task. The best recognition system reported in this work used a mixture of acoustic models like syllable, monosyllabic words and context dependent phones.

These previous studies have shown the utility of syllable as a subword unit. We have also seen that HMM models are capable of modelling temporal variations, and it is possible to incorporate constraints at various levels within the HMM framework. Local constraints of the language in terms of n-gram language models can also be taken into account while decoding. In the next section we look at some of the attempts to incorporate large span constraints into the language model.

2.4 APPROACHES TO MODELLING LARGE SPAN CONSTRAINTS USING LANGUAGE MODELS

The principal aim of language modelling is to characterize, capture and exploit the regularities in natural language. Statistical language models are widely used in most natural language applications. In statistical language modelling, a large text corpora is used to automatically determine the model parameters. Recalling the statistical

formulation of the speech recognition problem from (2.3)

$$W^* = \arg\max_W P(\mathbf{O}|W)P(W),$$

the estimate of $P(W)$ is obtained from the language model. Let $W = w_1, w_2, \ldots, w_n$ be the sequence of words that make up the hypothesised sentence. One way to estimate $P(W)$ is to use the chain rule

$$P(W) = \prod_{i=1}^{n} P(w_i|w_{i-1}, \ldots, w_1) \tag{2.5}$$

where $(w_{i-1}, w_{i-2} \ldots, w_1)$ is termed as the *history*. Since it may be difficult to compute $P(W)$ for large i, the length of the *history* is limited to $n-1$ in conventional n-gram models. For a trigram model (2.5) reduces to

$$P(w_i|w_{i-1}, \ldots, w_1) \approx P(w_i|w_{i-2}, w_{i-1}) \tag{2.6}$$

The performance of a language model is evaluated using a measure called perplexity [75]. The perplexity can be interpreted as the average branching factor of the language according to the model. It is a function of both the text and the model. When comparing perplexities of different models, the vocabulary of the model and the text used for testing must be the same. In such a situation a model with lower perplexity is supposed to be better. A good review of statistical language modelling techniques can be found in [1, 76].

The simplest of the statistical models are the n-gram models. An n-gram model uses the last $n-1$ words of the history to predict the next word. The larger the n, the higher is the differentiating power of the n-gram model. For a large n, due to the sparsity of data, the parameters of the model are poorly estimated. This reduces the

reliability of the prediction. For small n, the model is reliable, but its predictive power is limited. This is the trade off on n. the The advantages of the n-gram models are that they are easy to implement and interface with an application. They are good at capturing short term dependencies.

The main disadvantages of the statistical language models in general are [77]

1. They do not capture the semantics of the text.
2. Statistical models require a large amount of text, which may not always be available.
3. Statistical models often make no use of other linguistic or domain knowledge.

Despite the disadvantages, efforts have been made to extend the language modelling concept to model large span dependencies. The simplest extension of the n-gram approach to model large scale dependencies is to use higher order n-grams like 5-gram or higher. But, for a 5-gram, it is most likely that no sequence of $w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1}$ would have been seen in the training data. Hence the system needs to be backed off or interpolated with four-grams, trigrams, bigrams or even unigrams. However, it was shown in [1] that, for a large training corpus (about 250 million words), improvements in performance was observed for 5-gram, and even 6-gram models. The gain was obtained primarily due to improved smoothing techniques like interpolated Kneser-Ney smoothing. For a small training corpus, trigrams seemed to work best. The n-gram technique was extended up to a 20-gram model, and the improvement in performance tapered off after 6-gram model. But for most systems, going beyond trigrams is often impractical due to the trade off between memory and performance. As an extension of n-gram models, there are variable length n-gram [78–80] and X-gram models [81], where the goodness of the estimation of the probabilities is established by other criteria, which is not the length of the conditioning history.

When n-gram models with a large n are used, it is less likely that the context was seen before, but the chances of having seen a similar context will be high. Skipping models [77, 80, 82], use this information. For a 5-gram context, we could consider the contexts like $P(w_i|w_{i-4}, w_{i-3}, w_{i-2})$ or $P(w_i|w_{i-4}, w_{i-2}, w_{i-1})$. For example it is likely that we have never seen the phrase *"This is a Communist philosophy"*. A 5-gram predicting $P(philosophy|\, This\ is\ a\ Communist)$ would assign it a low probability. There are skipping models of the form $P(w_i|w_{i-4}, w_{i-3}, w_{i-2})$, which would assign a higher probability to $P(\text{philosophy}|\text{This is a } \ldots)$, since similar contexts like *"This is a socialist/capitalist/religious/good philosophy"* may have been observed in the training data. But the skipping models have only shown marginal improvements in performance over other approaches [1].

To reduce the number of parameters in the n-gram model and to improve the reliability of the parameter estimates, class-based n-gram models can be used, where the words can be clustered into classes. If C_i is the cluster to which word w_i is assigned, then a trigram class model in different forms could be one of the following:

$$\begin{aligned} P(w_3|w_1, w_2) &= P(w_3|C_3)P(C_3|w_1, w_2) \quad \text{or} \\ P(w_3|w_1, w_2) &= P(w_3|C_3)P(C_3|w_1, C_2) \quad \text{or} \\ P(w_3|w_1, w_2) &= P(w_3|C_3)P(C_3|C_1, C_2) \quad \text{or} \\ P(w_3|w_1, w_2) &= P(w_3|C_1, C_2) \end{aligned} \tag{2.7}$$

The quality of the model depends on the clustering technique. There exist automatic clustering procedures [83, 84] that can be used to derive the clusters, some of which are based on information theoretic criteria. This type of model gave good results for limited domains like ATIS (Air Traffic Information System) [85], or when manual clustering of the words was done [86]. For less constrained domains, the technique

does not seem to work so well. Decision trees and Classification And Regression Trees (CART) were also used to model long term history [87]. But this approach is highly computation intensive, and the gains (a 4% reduction in perplexity) over a trigram model was very small [76].

In the data corpus such as the Wall Street Corpus, there may be different sentence types like business sentences (information about many promotions, mergers, demotions), financial sentences (combining stock names, many numbers, stock market terminology) and general news sentences. There exists large span correlations among the words in a sentence (many numbers, promotions etc.,). To capture these constraints, sentence mixture models can be used [88]. In these models topic dependencies can be represented by using a sentence level mixture of m component models. The models constrain the topic to be consistent within a mixture component. Each mixture component can be identified with n-gram statistics of a specific topic or a broad class of sentences. The 'topics' can be determined by automatic clustering procedures. Here topic means any broad class of sentences that share a common subject matter or style. These models perform better than the conventional n-gram models, but involve a large computation cost in search, which may be reduced using a n-best re-scoring frame work.

To capture large span constraints, one could alternatively use trigger-based maximum entropy language models. The idea is that a word like *school* would increase its own probability as well as the probability of a similar word like *teacher*. Using these triggers, a 25% reduction in perplexity was reported in [77]. The difficulty with this approach is in determining the trigger pairs. Different pairs display markedly different behavior which limits the potential of low frequency triggers [89]. A variation of the maximum entropy approach is the whole sentence maximum entropy approach [90].

Here the probability of the whole sentence is predicted instead of individual words. The main difficulty with the whole sentence approach is that the training is complicated [91]. The benefits of the whole sentence model may be small when divided over all words [1].

If it is assumed that a suitable parser is available for the domain concerned, then the large span dependencies such as between a subject and its direct or indirect object can be taken into account. Such models are called structured language models or dependency language models. The syntactic information can be used to determine equivalence classes on the n-gram history [92, 93]. A good 11% and 24% reduction in perplexity over a baseline trigram model was reported in [94] and [95] respectively, for this approach. The main problem with this model is the reliance on the parser, and the assumption that the correct phrase will be assigned a high probability [96], which may not always be true.

High level semantic information can be used to incorporate large span constraints. The semantic information is diffused across the entire text under consideration. An approach that models the semantic information uses topic mixture models. In this approach, a *document* is defined as a set of semantically homogeneous sentences. Each document can be characterised by drawing from a large set of topics, usually predefined from a hand labelled hierarchy which covers the relevant semantic domain [88, 97]. The main disadvantage of this approach is the granularity of the clustering procedure [8]. One of the simpler approaches to incorporate large span semantic constraints and which tries to extend the word trigger concept is the use of latent semantic analysis for language modelling. This technique is discussed in the next section.

2.5 LATENT SEMANTIC ANALYSIS FOR MODELLING LARGE SPAN CONSTRAINTS

Latent semantic analysis is a theoretical method of extracting contextual-usage and meaning of words using statistical computations applied to large corpus of text [98]. The concept is that the aggregate of all contexts in which a word does or does not appear provides a set of mutual constraints that largely determine the meaning of words and sets of words with each other [5]. The LSA has been used as a tool to explain the acquisition and induction of knowledge in humans, as a source to estimate coherence of passage of texts and related areas in human cognitive phenomenon. The LSA uses a mathematical technique called Singular Value Decomposition (SVD), which is similar to factor analysis. The LSA is widely used in the field of indexing and retrieval, where it is referred to as Latent Semantic Indexing (LSI). The LSA has been adapted for language modelling. In this section we briefly review the work in latent semantic indexing and the use of LSA concepts to model large span constraints for language modelling.

Latent semantic analysis can be viewed as an approach to obtain approximate estimates of the contextual usage substitutability (semantic similarity based on context) of words in a large text segment. That is, large span semantic constraints among words based on their co-occurrence information can be captured by LSA. The LSA produces measures of word-word, word-document, document-document relationships, which are well correlated with human cognitive phenomenon. The similarity estimates derived by LSA are not simple contiguity frequencies, co-occurrence counts or correlations in usage. They depend on a powerful mathematical analysis that is capable of correctly inferring the deeper relationship that is not explicit but *latent*. This is the reason the approach is termed as latent semantic analysis. The similarities derived

by LSA between words and passages arise from the analysis of the text documents under consideration. It does not depend on any external information derived from the environment or experimental results obtained externally, or any other manual human interpolations. It makes no use of word order, and hence of syntactic relations, dictionaries, knowledge bases, semantic networks, grammars, syntactic parses or morphology. The LSA does not use the data being analysed as summed contiguous pairwise or tuple-wise co-occurrence of words. It considers the data as detailed patterns of co-occurrence of a very large number of words over a very large number of local meaning-bearing contexts, such as sentences or paragraphs treated as a whole. It ignores how the order of words produces the meaning of a sentence, but captures how the differences in word choice and differences in passage meanings are related. In the LSA approach the meaning of a word is represented as a kind of average of the meanings of all the documents in which it appears, and the meaning of a document as a kind of average of all the words it contains [5].

The latent semantic analysis is a technique that can look beyond local constraints or short span history. It was proposed as a fundamental computational theory of the acquisition and representation of knowledge [98]. It had been shown that the meaning similarities derived by LSA closely match those of humans. It was also observed that the rate of acquisition of such knowledge from text approximates that of humans. This was demonstrated on a variety of tasks based on human verbal concepts and synonym tests [99], in simulating subject-matter knowledge, predicting learning from texts, to explain the theory of acquisition, induction and representation of knowledge [98], as a computational basis of learning [100] and in other related areas [5]. These tasks of human cognition are not confined to a small window of words (short span constraints), but necessarily need to span the context under study. That is, large span constraints

are to be used. Use of LSA implies that large span constraints are being modelled implicitly. LSA has traditionally been explained in terms of statistical methods. A probabilistic interpretation of LSA can be found in [101]. The probabilistic LSA was introduced as a technique for the analysis of two-mode and co-occurrence data. In [102] latent semantic indexing was compared to statistical regression and Bayesian methods to explain its performance.

The LSA has been widely used in indexing and retrieval applications where it is termed as latent semantic indexing. The main deficiency in most indexing methods is that the words used for searching is not the same as those by which the information has been indexed. A user in different contexts, or with different needs, knowledge or linguistic habits, will describe the same information in different terms. It is unlikely there will be a good word to word match in the search. There is a need to capture the implicit structure and relations among the words used in a document. That is, the span under consideration is the entire document, or in other words, it is of large spans. Using the concepts of LSA it is possible to construct a semantic space of terms and documents such that documents that are closely associated are placed near one another. Thus in a retrieval query, the query is placed in the semantic space, and documents near that point are retrieved and returned. A good introduction to the LSI technique and relevant efforts on trying to bring out the relatedness among documents and words can be found in [4].

One of the earliest use of SVD for indexing and retrieval was reported in [103]. It was shown that SVD captures the implicit higher order structure in the association of terms and documents in a collection of texts. Many later experiments reported that LSI improved precision and recall rates in the information retrieval task [4, 104]. A query expansion interpretation of LSI and a normalisation technique to improve

precision rates was proposed in [105]. In the information retrieval task normally large volume of text data would be continually added to the database collection. Different approaches to add in new terms and documents into an existing LSI-generated database were discussed in [3]. Likewise, novel applications of the concept of LSA that include cross language retrieval, matching people instead of documents, and the utility of the approach in case of noisy input was also discussed. Instead of using LSI to index text documents, it was shown in [106] how LSI in combination with Self Organising Map (SOM) can be used to index spoken audio documents. The documents were represented as a vector of word counts, whose dimensionality was reduced by a technique called random mapping [107]. In spoken documents, the documents are short and the important words rare. To get meaningful distributions of the index words, smoothing by SOM is employed. The SOM also provides a way to visualise the results. A fast version of this procedure was reported in [106].

The effectiveness of the LSA depends on the SVD, and in turn, on the values arising out of SVD. The values arising out of SVD of the $term \times term$ co-occurrence matrix were first studied for the word sense disambiguation task in [108]. The normalised average (centroid) of the vectors of the words in a context (history) was used as an approximation of the semantic context. If at least some of the words in the context are frequently used to describe what the current context is about, then their vectors would pull the centroid towards the direction of that context or topic. This type of representation was found useful in the word sense disambiguation task. A theoretical framework for understanding the values in the reduced form of the $term \times term$ matrix using transitivity was given in [109, 110]. The improved performance of LSI over other approaches was due to its use of higher orders of co-occurrence information. It was shown that a connectivity path exists for every nonzero element in the truncated

$term \times term$ matrix computed by LSI. The higher order of co-occurrence among words influences the values in the truncated $term \times term$ matrix. Connectivity paths up to degree six were noticed when LSI was used in some standard document collections. It was shown that the value in the matrix at location (i, j) can be considered as the similarity between term i and term j. By extension, if it contains a negative value, then it represents the anti-similarity between term i and term j [111]. It was also shown in [112] that all the values in the $term \times dimension$ (order of truncation of SVD) matrix are not important for LSI. Removal of up to 70% of the values in the $term \times dimension$ matrix resulted in similar or improved retrieval performance (as compared to the standard LSI).

The use of latent semantic analysis for the task of developing language models that capture the large span constraints was first proposed in [6]. It is similar to word trigger based language modelling. The LSA provides a more comprehensive framework to handle trigger pairs across the entire document. Every word combination in the vocabulary that occurs in the document is viewed as a potential trigger combination. It is this approach that leads to the systematic integration of large span semantic constraints into the analysis [8]. The theoretical formulation, different smoothing techniques that can be employed on words and documents and the integration of the LSA language model with the standard n-gram model were discussed in [113]. The integration of the large span language model with a speech recognizer and consequent reduction in word error rates of the speech recognizer were discussed in [113]. Approaches to combine the n-gram model with the LSA model and estimate the LSA language model probabilities were described in [7].

The LSA language model does not make use of syntactic information, as it ignores the word order. A mathematical framework to incorporate the preceding syntactic

information into the model was proposed in [114]. This leads to a statistical model that uses the preceding syntactic model that uses the preceding syntactic information along with long distance semantic information to assign probabilities. This approach is called latent syntactic semantic analysis. Its performance is marginally poorer than the LSA model, but results in better probabilities than LSA for syntactic-semantically regular words. A probabilistic framework to simultaneously take into account the local word interactions (as in Markov chains), syntactic structure and semantic document information was proposed in the maximum entropy framework. This approach showed a significant (21%) reduction in perplexity over a baseline trigram model. In the LSA based language models, as the history of the document increases, it may sometimes contain irrelevant information for predicting the next word. The history can be discounted as in [115]. In another approach, the history was partitioned into three levels corresponding to document, paragraph and sentence levels [116]. Information derived from these three levels was combined using a soft max network to obtain improvement in the perplexity over a trigram baseline.

It is difficult for the LSA language model to capture the large span word dependencies at the beginning of a document. Due to the shortness of the history, in [117], the word to be predicted was treated as an LSI query, and the closest matching document was retrieved. This was used as the history for the word. Further, the LSA-based parameter smoothing, and ways to determine interpolation coefficients for the language models was also suggested. This approach reduced the perplexity by 49% when compared to a bigram model.

In this section we have seen how LSA initially proposed for indexing and retrieval task was also shown to model human cognitive behavior. This was due to the ability of the model to look at large passages/documents and derive the latent constraints

among the words. This model was adapted for language modelling due to its ability to model large spans. In the next section we see how large span constraints can be used to improve speaker recognition systems.

2.6 APPROACHES USING LARGE SPAN CONSTRAINTS IN SPEAKER RECOGNITION

Humans are able to recognize a person by his voice from among a familiar set of speakers. Humans can use knowledge right from the speech segment level to discourse level depending on the requirement to come up with a decision on the identity of the speaker. A wide variety of approaches to automatic speaker recognition by machine exist [118, 119], but very few approaches explore large span constraints for the task.

Voices of two persons differ due to the physical differences in the vocal organs and the manner in which they use them during speech production. The characteristics of the vocal chords are reflected in the pitch information. It was shown in [120] that pitch contour (a suprasegmental large span feature) has some advantages over spectral information for speaker recognition. Pitch contours of the entire sentence provide useful information to distinguish between speakers, as it is unlikely that an impostor can mimic the entire variation of pitch as a function of time. Pointers to several studies on the acoustic features suitable for speaker recognition was reported in [121]. They found that long term parameter averaging of pitch, gain and reflection coefficients for up to 1000 frames was shown to improve the between-to-within speaker variance ratio. Other attempts to model the pitch contour to capture speaker-specific information were also successful. Pitch, accent features and word durations were used as features to train multiple neural network to capture speaker specific information. The evidence was combined to arrive at a decision for text dependent speaker verification task [122].

In another related work piecewise linear model is fitted on the pitch contour to obtain a stylized pitch contour. Combining this evidence with a Gaussian Mixture Model (GMM) based system for speaker recognition improved the performance of the system.

Traditionally work on speaker recognition has focused on characterising the statistics of the speakers amplitude spectrum. Humans are able to distinguish between speakers who are familiar to them far better than those who are unfamiliar. They are able to identify a speaker easily based on his idiosyncrasies like indicator words (e.g., *I, OK, yeah, uh-uh, right*), conversational-style features such as pause and turn lengths, discourse markers (e.g., *you know*), back channel expressions (e.g., *all right, sure*), editing markers (e.g., *I mean*) [123, 124]. These idiolectic characteristics are spread over the entire conversations (large spans), and need to be modelled. Extensive literature on similar work based on authorship attribution is available for determining the authorship of disputed/unclaimed text based on samples of text by target authors. We first briefly review the work in this area since a similar approach is adopted in this work for speaker recognition.

The task of authorship attribution is to determine the identity of the author of anonymous or doubtful text, given some prior stylistic characteristics of the author's writing, extracted from a corpus of his known works. Here the text is derived from a speech recognizer. The first studies in 1887, 1901 on stylometry was based on word-length distribution [125]. A naive Bayes classifier approach was used to determine the authorship of the *Federalist papers* [126]. In this approach, the independence of function words usage was assumed, which may not be true. A Support Vector Machine (SVM) based approach to identify the authors of the *Federalist papers* was reported in [127]. An approach based on style markers derived from multiple parses of a sentence was proposed for the Greek language [128]. The approach avoided the use of lexical

based measures such as word frequency, and can be used even when the training data is small. A character level n-gram language model based approach which uses Bayesian decision theory for identification was proposed in [129]. The reported advantage of the approach is that minimal preprocessing and feature selection was involved.

Just as an author is identified by his writing style, the subject expertise and areas of his interest may be derived from his previous written works. The tedious task of routing papers for review in conferences based on the reviewers' expertise and choice of review areas was automated based on the principles of information retrieval and latent semantic indexing [130, 131]. The reviewers' interests were represented based on abstracts of the papers written by them (i.e., treated as documents for LSA). The paper to be assigned was treated as a query, and the closest reviewer based on his abstracts (documents) was chosen for routing the paper. The performance of the system was found to be fairly reasonable when compared to the manual assignment. For speaker recognition we adopt an approach to derive the identity of the speaker from the words spoken by him, in which his idiolectic characteristics are embedded.

Each speaker has a set of idiosyncrasies like usage of certain words or phrases, intonation, stress and timing of the words. These patterns of speech manifest in conversational speech over large spans. A large volume of conversational speech of a speaker, and the transcription of the speech, were available in the NIST extended data speaker detection task [118]. This was the motivation to capture the speaker idiosyncrasies from text corpora [132]. From the text transcription of the conversations, the word unigrams and bigrams for the corpus was found. Using the target/background likelihood ratio framework, it was shown that the idiolectic characteristics can be captured. In this framework the performance of the speaker recognition system was reported to be the highest when the most frequent bigrams were used. Thus it was suggested

that commonly used word patterns have speaker discrimination power. The Stanford Research Institute (SRI) database of prosodic features for the Switchboard-I corpus includes a wide range of prosodic, lexical and disfluency information. In [124] the pitch related features, duration related features, indicator word usage, conversational style features, pause fillers, discourse markers, back channel expressions and other such features were used in conjunction with a Gaussian Mixture Model GMM and Language Model (LM) based speaker recognition systems. The individual GMM and LM speaker recognition system performance improved when these large span prosodic features were included. The best performance was obtained when all the three systems (GMM, LM and prosodic) were combined using interpolation.

In this work, we try to model the speaker idiosyncrasies that can be derived by using n-gram terms in the LSA framework. This approach provides complementary information that can be combined with other approaches for speaker recognition.

2.7 SUMMARY

In this chapter we reviewed the work related to selection of appropriate units for speech recognition, and discussed how these units could be modelled acoustically. The constraints that can be imposed on the recognizer in terms of a language model and the usefulness of large span constraints in a language model were reviewed. The usefulness of these constraints for speaker recognition and approaches to incorporate large span constraints into a speaker detection system were also reviewed. In the next chapter we discuss the development of a speech database for Indian languages, select appropriate subword units for speech recognition in Indian languages, and study the statistical characteristics of the words and subword units from the perspective of developing speech systems.

CHAPTER 3

ANALYSIS OF SPEECH DATABASE

3.1 INTRODUCTION

Speech research is geared towards providing a spoken language interface to converse freely with a machine. Spoken language input to a machine may involve different technologies like language identification, speaker recognition, speech recognition, natural language understanding, dialogue management and concatenative text-to-speech synthesis to varying extents. In this study our focus is on speech recognition. Speech recognition involves transformation of the input speech signal into a sequence of symbols (units). The sequence of symbols is then converted into a sequence of words of the target language corresponding to the message in the speech signal using linguistic knowledge. The large variability in the speech signal makes speech recognition a difficult task. The main sources of variability can be categorised into acoustic variability, inter-speaker variability and intra-speaker variability.

1. The acoustic variability could be due to: (a) the different realisations of a sound (phoneme) in different contexts, termed as phonetic variability, (b) the changes in the acoustic environment (e.g. people speak differently in noisy and quiet environments) and (c) the differences in the channel or transducer type.

2. Certain wide variations can also be caused in the speech signal due to changes in the speaker's physical and emotional state, speaking rate and elocution mode (read speech, spontaneous speech or conversational speech). These can be termed as intra-speaker variability.

3. Inter-speaker variability may occur due to the socio-linguistic background of the speaker, speaker-specific mannerisms (idiolectic traits), the dialect spoken, and due to the shape and size of the speaker's vocal tract.

A large annotated speech database, structured and organised to address the issues of variability, is required to develop robust models for speech recognition. The task of generating and annotating a large speech corpus is labour intensive and expensive. To address the diverse speech tasks (speech recognition, speaker recognition, etc), it is desirable to have a speech corpus addressing each specific task. This is preferable, as the type of data required for each task is different. This would increase the effort involved in data collection many fold. For Indian languages the need for an annotated database, suitable for use in speech recognition, speaker recognition, text-to-speech synthesis, language identification and information retrieval was felt. Towards this goal we developed the Indian Institute of Technology Madras (IITM) speech corpus [133], the design and development of which is described in Section 3.2.

The vocal tract is unique for every speaker, like fingerprints. Sounds produced by it are also distinct. But there is a commonality at the perceptual level that is useful in identifying the basic sounds in a language. Phonemes are the basic sound units in a language that are perceptible, and are useful to distinguish between words. The number of phonemes in a language is usually small. The phonemes have been the choice as the subword unit for most speech recognizers. The speech databases are thus usually annotated in terms of phonemes for speech recognition purposes.

To develop robust acoustic models for subword units in a speech recognizer, knowledge of the acoustic-phonetics of the basic speech sounds in different contexts and environments is desirable. A thorough study of the characteristics of the subword unit at the segmental level and their interactions at the suprasegmental level is desirable.

To facilitate our understanding of the speech sounds and the issues involved in the development of a speech recognizer, we studied some of the acoustic, prosodic, statistical and information theoretic aspects of the sound units in three Indian languages, namely, Tamil, Telugu and Hindi. The details of the studies are presented in the following sections.

Previous studies on the phonetic properties of the sound units in five European languages suggested a cohort (equivalent class) type classification of the sound units. These cohorts could then be used as the basic classes for a preliminary recognizer [134]. Properties of words in large lexicons and their structural properties with implication to isolated word recognition was discussed in [135]. In [136] a study of the duration of the vowels and consonants in Telugu in different contexts, and its usefulness for text-to-speech synthesis and speech recognition was described. Studies on the statistical properties of words and syllables in text corpora of Indian languages was reported in [137, 138]. These previous studies throw light on the characteristics of the sound units in a language which may be used for the development of a speech recognizer.

The following sections deal with the study of the statistical properties of words and subword units suitable for speech recognition in Indian languages. Observations on the relative frequency of the sound units, the structural properties of words and subword units and their durations are presented. The consonants can be grouped based on their acoustic-phonetic descriptions. Observing the transitions of the consonant sounds between the different groups, rules can be derived to constrain the search space of a speech recognizer. Rules defining the permissible and prohibited consonant cluster transitions in a language can also be derived. These rules may be useful in language identification systems.

3.2 DATABASE DESIGN AND DEVELOPMENT

To develop speech systems in Indian languages, a large speech and text corpora that cover the domain of interest are required. It is preferable that the corpus conform to the requirements of multiple speech systems. The corpus should be uniform across all the languages. Uniformity here refers to the amount of text and audio per language, quality of data such as recording conditions (noise, channel, microphone etc.), collection scenario (task, setup, speaking style etc.) and transcription conventions [139]. This uniformity would enable a comparison of results in a task across languages.

3.2.1 Issues in database design

In addition to the inter-speaker, intra-speaker and contextual variability that need to be addressed during database design, the following issues also need consideration [140]:

1. The quality of the database (quality of the audio signal and annotations).
2. The content of the database (items to record, sentences, numbers, isolated words/syllables, spontaneous speech).
3. Quality of speakers used for recording.
4. Cost of collection of data (renumeration for participation, annotation, supervision).
5. Validation cost (validation by humans).
6. Development cost (cost of hardware and software).

3.2.2 Corpus development

For developing speech systems in Indian languages, we need a database suitable for multiple tasks described earlier. A database of clean speech is preferred over telephone speech so as to reduce the channel effects. Due to resource constraints, read speech

was preferred over spontaneous speech. Also, it is faster and less labour intensive to process read speech than spontaneous speech. The speech recorded should have minimal disfluencies. It was observed that not everybody is capable of reading text aloud with minimal errors [141]. The spoken realisation of a word depends on the accent used by the speaker. Further, journalistic type of writing is lengthy and is more difficult to read for recording purpose by the general population. It is also preferable if the text selected for reading is of the same genre as the text corpora used for language modelling, as it helps improve speech recognition performance. Considering these issues, our preferred choice is to use TV news bulletins. The TV news is read speech and the speakers are trained to read aloud with minimal disfluencies. Generally the accent of the TV news readers is neutralised. The speech recorded after transmission is close to clean speech, and can be obtained using a good quality recording equipment. As a large text corpus is required for language modelling, online newspaper archives can be used, whose genre would be similar to those of the news bulletins. For text-to-speech synthesis this read speech would be useful to derive durations of the base units [142]. The news bulletin speech corpora can also be used for information retrieval, and for close-set speaker recognition studies.

3.2.3 Speech data collection

TV news bulletins broadcast by Doordarshan in 3 Indian languages Tamil, Telugu and Hindi are recorded during the same period. The goal is to record similar news content in all the languages. This would facilitate language independent information retrieval studies where the semantic content is similar. For man-machine dialogue systems and information retrieval systems, names of people and places form an important part of the dialogue items/search terms. A database with similar news content would contain a large number of examples of these terms, enabling development of language indepen-

dent audio content analysis and dialogue systems. The TV broadcast is recorded on a high quality VHS video tape. The audio signal from the tape is sampled at 16 kHz with 16 bit resolution using a high quality 'sound blaster' sound card on a personal computer, and stored in standard wave format. Each news bulletin comprises of about 10 to 15 minutes of read speech. The total duration of the digitised audio in each language is about 5 hours.

3.2.4 Speech transcription

For speech recognition, the transcription of the spoken sentences should represent what is actually spoken rather than the grammatically correct version of the sentence. This would enable developing better acoustic models. To facilitate transcription, the read speech is segmented into short segments of approximately 3 sec duration, containing only sentences spoken by the news reader. Speech by all other speakers in the news bulletins are ignored, since the speaking style, dialect and speaking rates are likely to vary widely. Speech utterances corrupted by channel effects or other disturbances are removed. The speech segments were then orthographically transcribed manually into the common Indian language TRANSliteration (ITRANS) code for Indian languages, which uses the Roman script [143]. The ITRANS code is chosen, as it used the same symbol to represent common sounds across Indian languages. This forms a good intermediate script to compare, contrast and represent the sound units of different Indian languages. The transcribed sentences are parsed into syllables based on the rules of the language. The speech segments are then segmented into syllables manually by trained persons. During transcription, the mispronounced phrases are removed. For words with multiple pronunciation, the transcription is based on the actual manner in which the word is spoken in the utterance. The transcription and annotation has been verified by at least two people. Errors in the transcription have been corrected.

Any misalignments in the marking of the boundaries of syllables have been realigned. The database is organised in a TIMIT-like format [144]. Each news bulletin in the speech database is organised into separate directories. The original wave files, the transcription of the speech files and time aligned segmentation of the speech file in terms of words and syllables are contained in four separate subdirectories. The Tamil database has 33 news bulletins, of which 10 are spoken by males and 23 by females. The Telugu database has 20 news bulletins, of which 11 are spoken by males and 9 by females. Similarly for Hindi, of the 19 bulletins 6 are spoken by males and 13 by females. The statistics of the database is given in Table 3.1.

Table 3.1 Description of Indian languages database.

Language	Total duration (min.)	# bulletins		# speech segments	Total # words	# distinct words	Total # syllables	# distinct syllables
		Male	Female					
Tamil	242.1	10	23	7,359	30,688	8,947	100,707	1,975
Telugu	219.7	11	09	6,484	25,463	9,218	84,349	2,273
Hindi	139.1	06	13	4,191	26,090	5,162	50,237	2,002

3.2.5 Database used for studies in this thesis

For the purpose of developing speech systems, the characteristics of the sound units in three Indian languages, Tamil, Telugu and Hindi are studied. To study the statistical properties of various subword units from an information theoretic perspective, the transcribed text of speech in the Tamil language is used as an illustration. The viability of developing syllable-based speech recognition systems using the framework of HMM is illustrated with the Tamil and Telugu languages. The development of text-based and speech-based large span constraint models both at the word level and the syllable level are illustrated with the Tamil language. The concept of speech-based large span

constraint models is language independent and can be extended to any number of languages. In general, all the above mentioned studies can be extended to other languages.

3.3 STATISTICAL ANALYSIS OF THE SPEECH DATABASE

Each language has different types and number of basic speech units, different pronunciations, syntactic, semantic and pragmatic constraints which affect the production of the speech signal. Most Indian languages are syllabic in nature. A syllable has an obligatory vowel (V) which forms the nucleus. It may be preceded by one or more consonants (C) in the onset. The coda consist of the consonants following the vowel. In general the syllabic units are of the form C^mVC^n, where $m, n \geq 0$. The total number of such syllabic units in a language is large, typically more than 5,000. The statistical characteristics of these syllabic units is described in the following sections. The statistics presented in the following section pertain to the speech corpus described in Table 3.1. The inferences are provided from speech recognition point of view.

3.3.1 Frequency of occurrence of syllabic units

A consonant in speech is generally pronounced along with a vowel, either preceding or succeeding it (as CV or VC). In most Indian languages, most characters are represented as a single CV type unit. Another characteristic of Indian languages is that most characters are written in the same manner as they are spoken, i.e., they are phonetic in nature. The languages of India have a common phonetic base. The syllables are built from the basic set of sounds represented by the vowels and consonants of the language. The vowels number between thirteen and eighteen, while the consonants vary from eighteen in Tamil to as many as thirty eight in Telugu and Malayalam. The basic CV part of every syllable, ignoring the multiple consonants if any occurring in

the onset or coda, is considered for computation of the statistics shown in Table 3.2. As an illustration, the frequency counts for the Telugu language is shown. From the Table it is observed that the CV units with long vowels, are less frequent than the corresponding CV units with short vowels. The exception being CV units containing the long vowel /E/. Similarly the aspirated CV units are less frequent compared to the unaspirated CV units. This is in tune with Zipf's principle of least effort which states that an individual behaves in such a way that he tends to minimize the work he has to do [145]. More effort is required to pronounce the aspirated sounds and those with long vowels. Hence the aspirated CV units and the CV units containing long vowels used in spoken language are lesser in number. In Telugu it is also seen that, CV units with semivowels /r/ and /l/ are most frequent, followed by CV units with nasal consonants in Telugu. If the most frequently occurring syllables in a language are considered, it is observed that a small subset of the syllables in the database cover about 75%to 85% occurrences of the syllables in the database. Table 3.3 shows the number of most frequently occurring syllables required for a specific coverage of the database. From Table 3.3 one can conclude that if such a small subset of the syllable vocabulary is modelled efficiently in a speech recognizer, a majority of the syllables in spoken utterances can be recognized.

Table 3.2 Number of occurrence of different Vowels and CV classes in Telugu expressed as percentage.

	a	A	i	I	u	U	e	E	ai	o	O	au
	1.6355	0.8824	0.5894	0.7899	0.5029	0.0083	0.7389	0.2467	0.1459	0.2277	0.0356	0.0024
k	2.0684	1.2299	0.7128	0.0806	1.2773	0.1447	0.1091	0.2989	0.0178	0.3416	0.3190	0.0202
kh	0.3843	0.0593	0.0059	0.0036	0.0083	0	0	0	0.0071	0	0.0012	0
g	0.7958	0.7602	0.4447	0.0557	0.5171	0.0130	0.0119	0.0332	0.0083	0.0937	0.0486	0.0474
gh	0.1791	0.0451	0.0119	0.0036	0.0036	0	0	0	0	0	0.0119	0
∼N	0.0083	0.0012	0	0	0	0	0	0	0	0	0	0
ch	0.6226	0.5266	0.8124	0.0237	0.1174	0.0119	0.3463	0.8337	0.0285	0.1139	0.0047	0.0285
chh	0.0036	0.0036	0.0012	0.0024	0	0	0	0.0036	0	0	0	0
j	0.8883	0.2087	0.1803	0.1032	0.3771	0.0190	0.0676	0.1554	0.0083	0	0.0166	0.0012
jh	0	0	0	0	0	0	0	0	0	0	0	0
∼n	0.0202	0.0036	0	0	0.0024	0	0	0	0	0	0	0
T	0.8729	0.2751	0.6499	0.4625	0.5029	0.0178	0.1115	0.0593	0.0296	0.0059	0.0273	0.0012
Th	0.0344	0.0047	0.0059	0	0	0	0	0	0	0	0	0
D	0.3653	0.3143	0.5242	0.0735	0.9298	0.0273	0.0510	0.0296	0.0451	0.0107	0.0308	0
Dh	0.0036	0.0178	0.0700	0.0047	0	0	0	0	0	0	0	0
N	0.4222	0.1340	0.0889	0.0083	0.0166	0	0	0	0	0.0059	0.1174	0
t	1.7434	0.6808	0.7270	0.3143	0.7009	0.1909	0.3333	0.1044	0.0036	0.1162	0.4341	0.0024
th	0.1471	0.1364	0.0700	0.0024	0.0154	0.0036	0.0024	0	0	0	0	0
d	1.5714	0.3487	1.1172	0.0996	0.6037	0.0439	0.0285	0.3926	0.0249	0.0178	0.0463	0.0047
dh	0.1992	0.3095	0.4032	0.0700	0.0806	0.0012	0.0047	0	0.0024	0.0024	0	0
n	3.1749	1.2726	**3.3860**	0.1328	1.0591	0.0522	0.0961	0.1909	0.0273	0.0012	0.0320	0.0047
p	1.4018	1.4516	0.3973	0.1589	0.3997	0.0901	0.1186	0.1233	0.2787	0.1233	0.2455	0.0166
ph	0.0605	0.0249	0.0142	0.0095	0.0130	0.0024	0.0320	0	0.0356	0.0012	0.0178	0.0024
b	0.3902	0.5456	0.1743	0.1222	0.1115	0.0095	0.0463	0.0119	0.0237	0.0083	0.0427	0
bh	0.3439	0.3902	0.1388	0.0142	0.1886	0.0427	0	0.0036	0.1067	0.0012	0.0083	0.0095
m	1.8822	0.8148	0.4507	0.1826	0.5230	0.1696	0.1281	0.0795	0.1506	0.0866	0.0818	0.0095
y	2.8808	0.7780	0.5218	0.0059	0.2574	0.1174	0.0498	0.0901	0.0095	0.0178	0.1257	0
r	**4.5151**	1.8442	1.7090	1.0603	2.2700	0.1222	0.5977	0.1767	0.1020	0.0296	0.5195	0.0047
R	0	0	0.0024	0	0	0	0	0	0	0	0	0
l	**3.4026**	0.4210	0.6582	0.1803	1.7434	0.0415	0.0403	0.2455	0.0463	0.0249	1.9391	0.0036
L	0.0961	0.0925	0.0463	0.0059	0.0178	0.0047	0	0.0059	0.0012	0	0	0
v	1.8146	0.9251	1.0875	0.1210	0.2230	0.0083	0.1637	0.3807	0.2704	0.0249	0.0249	0
sh	0.4554	0.5420	0.1886	0.0415	0.0415	0	0.0036	0.0415	0.0047	0	0.0142	0
Sh	0.3534	0.0510	0.0996	0.0178	0.1198	0	0.0225	0.0190	0	0	0.0036	0
s	1.8027	0.5657	0.6843	0.1720	0.3582	0.0380	0.2467	0.0854	0.1660	0.0107	0.0451	0.0142
h	0.3143	0.2443	0.1898	0.0071	0.0225	0.0012	0.0510	0.0308	0.2040	0.0012	0.0344	0.0012
kSh	0.0059	0.0024	0	0	0	0	0.0142	0.0154	0	0	0.0036	0

Table 3.3 Number of frequently occurring distinct syllables required for a specific coverage of the database.

Language	# distinct syllables	Coverage (%)		
		85	80	75
Tamil	1,975	301	230	177
Telugu	2,273	278	207	160
Hindi	2,002	465	326	239

3.3.2 Word and syllable patterns

Among the words in the database, the number of vowels in a word varies from 1 to 12 in Tamil and Telugu, and between 1 to 10 in the Hindi language. The average number of vowels per word and the consonant to vowel ratio is listed in Table 3.4. The average number of vowels per word in Telugu is high. This correlates with the known phonetic knowledge of the language (most words in Telugu end with a vowel). Since monosyllabic words are more frequent in Hindi, the average number of vowels per word is lower in Hindi. Due to the syllabic nature of Indian languages, and the predominance of CV and CVC type of syllables (discussed in the next section), it is seen from Table 3.4 that the C/V ratio is smaller for the three languages studied when compared to the European languages French, English, Swedish and German, which have a C/V ratio of 1.35, 1.41, 1.58 and 1.71, respectively [134]. The C/V ratio is indicative of the larger consonant clusters that occur in the words of a particular language. This suggests that Indian languages have smaller consonant clusters.

Table 3.4 Average number of vowels per word and consonant to vowel ratio for the three languages.

Language	Average number of vowels/word	C/V ratio
Tamil	3.28	1.31
Telugu	3.31	1.26
Hindi	1.93	1.28

3.3.2.1 Structural patterns of syllables

The study of the distribution of the sound units and the constraints among the sound sequences is useful in constraining the search space and correcting the errors in recognition [135]. The structural patterns of the syllables occurring in the database are studied. For the syllable vocabulary in a language, the different C^mVC^n structural patterns are obtained by replacing the consonants in the syllables by (C) and the vowels by (V). The patterns thus obtained are listed in Table 3.5.

Each type of structural pattern can be considered as an equivalent class. The set of syllables that map onto an equivalent class are called 'cohorts'. From Table 3.5 we see that the syllables that occur most frequently are of CV type, followed by the syllables of CVC type in all the three languages. In combination, these two types of syllables cover around 89% of the frequently spoken syllables in a language. For better performance of a syllable recognizer, accurate modelling of these CV and CVC type of syllables would be desirable. If a broad phonetic preclassification/analysis of the speech input in terms of these syllable patterns (equivalent classes) is conducted, it would vastly reduce the size of the syllable lexicon to the number of cohorts in that

Table 3.5 Frequency of occurrence of syllable patterns.

Type of syllable patterns	Frequency of occurrence (%)		
	Tamil	Telugu	Hindi
CCCV	0.004	0.002	0.004
CCCVC	-	0.002	0.002
CCV	0.81	3.79	2.65
CCVC	0.46	1.10	0.83
CCVCC	0.17	0.03	0.09
CV	56.56	65.70	62.63
CVC	33.94	23.04	24.80
CVCC	0.11	0.47	1.35
CVCCC	0.003	-	0.02
V	4.57	3.43	3.65
VC	3.50	2.42	3.83
VCC	0.006	0.01	0.15
VCCC	-	0.001	0.004

equivalent class [135]. Consequently, the search space is reduced and this improves the overall recognition performance. In [25] such a preliminary classification into stop consonant vowels and nonstop consonant vowels prior to recognition of the syllables yielded a better recognition rate. We show in Section 4.6 approaches to preliminary classification of these syllable patterns and their utility for speech recognition [146].

The number of syllables per word in each of the three languages and their coverage of the database is listed in Table 3.6. Among the three languages Hindi seems to have the most number of monosyllabic words. Tamil has more bisyllabic words, while Telugu has maximum number of trisyllabic words. The coverage by these multisyllabic words seems to suggests that bigram or trigram language models at the syllable level may perform better as, within word syllable sequence constraints would be modelled better.

3.3.2.2 Structural patterns of words

Similar to syllable structural patterns, the word patterns are determined by converting all the words in the database into a sequence of C's and V's. It is observed that there

Table 3.6 Coverage of database by multisyllabic words.

# syllables per word	Coverage of database (%)		
	Tamil	Telugu	Hindi
1	4.73	8.85	32.72
2	31.20	22.41	23.76
3	28.04	28.22	23.26
4	17.43	21.60	11.04
5	10.26	11.00	5.91
6	4.33	5.12	2.11
7	2.69	1.80	1.00
8	0.90	0.63	0.16
9	0.22	0.15	0.04
10	0.13	0.09	-
11	0.03	0.03	-
12	0.01	0.01	-

are more than 1,100 different Structural Patterns of Words (SPW) in all the three languages together. Among these structural patterns, there exist some SPW that occur frequently in one language and do not occur in the other languages. These unique SPW may be useful in distinguishing among a small set of languages.

Observing the most frequently occurring words (those with at least 500 occurrences) that have the same SPW in the database, we see from Table 3.7 that most of these word patterns have less than 5 syllables in them. The duration of most of these words are all less than 450 msec. The more frequently used words have fewer syllables, and are of smaller duration, which is again in tune with Zipf's principle of least effort.

Table 3.7 Most frequently occurring structural word patterns with the number of occurrences greater than 500, along with their average durations for the three languages.

Tamil			Telugu			Hindi		
Word pattern	# occurrences in transcripts	Avg. duration msec.	Word pattern	# occurrences in transcripts	Avg. duration msec.	Word pattern	# occurrences in transcripts	Avg. duration msec.
CVCV	2117	229.9	CVCVCV	1849	371.7	CV	5472	150.1
CVCCV	1572	314.4	CVCCV	1509	314.6	CVC	3010	196.3
CVCCVC	1526	361.6	CVCV	1413	279.9	CVCV	2777	269.3
VCCV	1522	277.7	CVCCVCV	1208	398.5	CVCVC	1837	324.7
CVCVCV	1256	368.2	CVCVCVCV	1195	435.4	CVCCV	1399	328.7
CVCVCVC	1098	411.1	VCVCV	700	314.6	VC	1247	164.4
CVCVC	1050	298.6	CCV	637	221.4	CVCCVC	926	391.3
CVCVCCV	895	362.4	CVCCVC	551	367.6	CVCVCV	874	388.1
CVCCVCVC	842	448.0	CV	527	166.9	CVCVCVC	609	422.3
CV	634	168.6	CVCVC	514	329.8			
CVCCVCV	616	406.3	CVCCVCVCV	514	541.1			
CVCVCVCVC	603	491.6	CVCVCCV	512	426.2			
VCCVCV	589	389.2						
CVCVCVCV	575	458.9						
VCV	514	244.1						

3.3.3 Common sound units across languages

For multi-lingual speech recognition, one basic approach is to integrate several mono-lingual recognizers with a front end language identification system [147]. Another approach is to develop models by pooling data from all the languages for similar sound units. The number of syllables common between any two languages and those common across all three languages are shown in Table 3.8. For these syllables, data can be pooled together for developing acoustic models. Telugu and Hindi have aspirated sounds, while Tamil does not have them. Due to this reason, though Tamil and Telugu belong to the same family of languages unlike Hindi, Telugu and Hindi have more syllables in common. Tamil and Hindi have the least number of common syllables. Analysing the individual frequency of occurrence of the 540 syllables commonly occurring across all the three languages, it is observed that only 192 of them have more than 30 examples in each of the languages. This indicates that only a subset of the common syllables may have sufficient examples to train acoustic models of speech syllables.

Table 3.8 Number of syllables common across languages.

Language combination	# common syllables
Tamil - Telugu	834
Telugu - Hindi	992
Hindi - Tamil	636
Tamil - Telugu - Hindi	540

3.3.4 Duration of syllables

In text-to-speech systems, knowledge of the duration of the basic sound units is important. Duration knowledge can be used to incorporate constraints at the local and global level. At the level of subword units, duration knowledge provides local constraints that can be used in acoustic modelling of the sound units. At the word and phrase level, duration knowledge forms part of the prosody constraints that operate at the global level, and can be used in most speech-based applications. Incorporation of durational constraints in HMM based recognizers marginally improves the performance of the system. In this section some aspects of the durations of the syllabic units are studied. The average duration of the syllables in the three languages is plotted as a histogram in Fig. 3.1. The range of durations of the syllables vary from 50 to 420

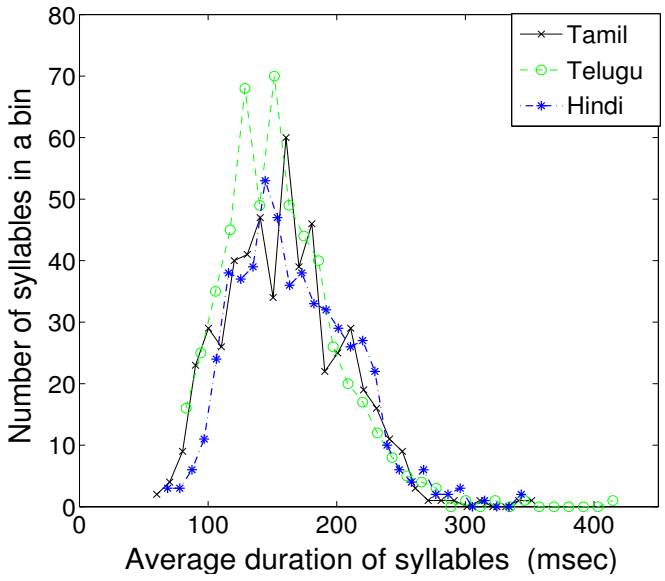

Fig. 3.1 Histogram of average duration of the syllables in three languages.

msec. The average duration of the syllables is around 140 to 160, msec in the three languages.

In each of the three languages, the average duration of the frequently occurring

syllables (with at least 100 occurrences) in the database is computed. There are 194, 166, and 100 such syllables in Tamil, Telugu and Hindi, respectively. The average durations of this subset of frequently used syllables is 173.3, 165.3 and 190.8 msec for Tamil, Telugu and Hindi, respectively. The scatter plots of the average durations of the syllables is shown in Fig. 3.2. It is seen from the figure that the average duration of the most frequently occurring syllables is less than the overall average durations mentioned above. This is again in tune with Zipf's principle of least effort, that an individual tries to minimize the work he has to do by using smaller durations for the most frequently used syllables. The average duration of the syllables in Hindi is larger than that of the other two languages.

3.3.5 Consonant-consonant transition

The consonants in a language can be broadly grouped into velar, retroflex, dental, bilabial, nasal, semivowel, fricative, affricate, flap, lateral and trill classes. Study of the pattern of occurrence of these consonant combinations provides information about the permissible consonant combinations and their occurrence in word initial, medial or final positions in a language. Transition probabilities between a pair of consonant groups in each of the three languages is given in Tables 3.9, 3.10 and 3.11. The symbol $< /s >$ in the table denotes sentence end. The study is limited to transitions between a pair of consonants due to the difficulty in interpretation of transition probabilities for higher orders. The rules derived from the transition probabilities can act as pointers for language identification systems, and in understanding the local constraints that govern consonant clusters in a language.

From the tables it is observed that in Tamil velar to velar and retroflex to retroflex transition probabilities are the highest. These transition probabilities are highest for any consonant combinations among all the three languages. Another feature in Tamil

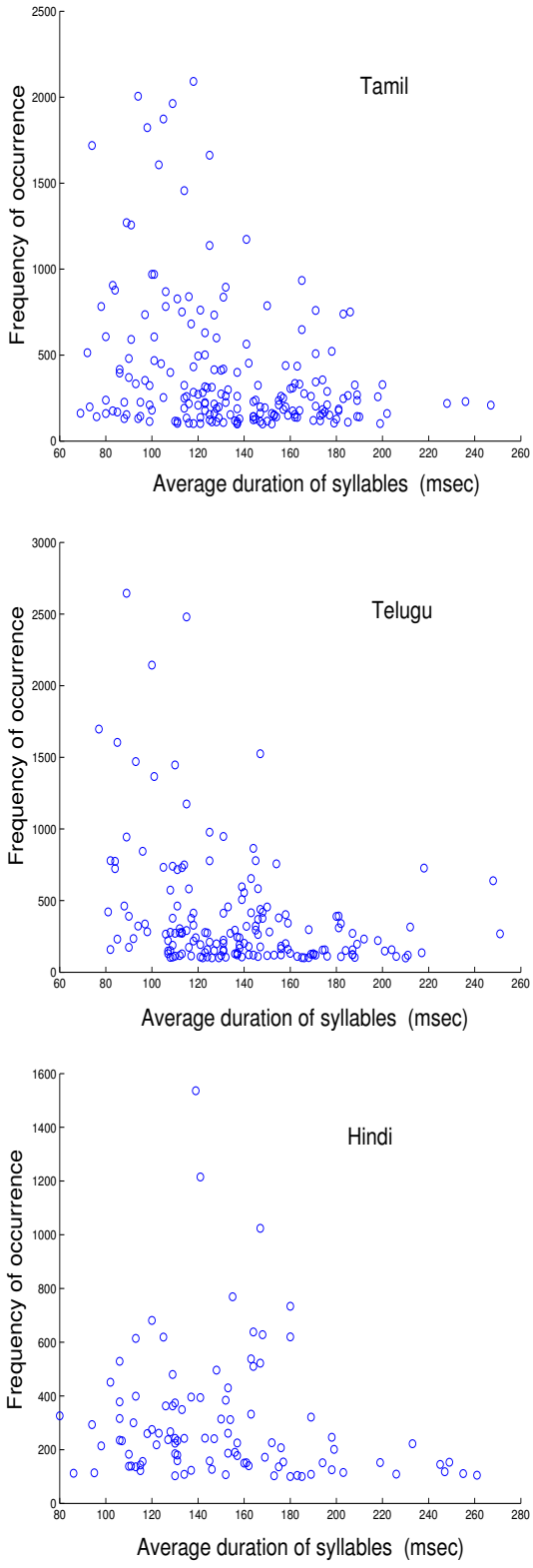

Fig. 3.2 Scatter plots of most frequently occurring syllables in three languages.

is that the transition probability among the consonant geminates is highest. Such high frequency of occurrence of geminates is not observed in the other languages. Likewise, the probability of transition from a sound unit in any other consonant group to a sound unit in the dental group is much higher in Tamil than in the other two languages. Though the consonant geminates are very frequent in Tamil, the flap-flap geminate is not observed in this study. As the database is small (around 25,000 to 30,000 words), it is not possible to conclusively comment that certain inter-group transitions are impossible/excluded in a language. But a study of the transitions between consonant groups gives us an important insight into the structure of the language. Rules derived from a study of these transition probabilities may be useful as additional knowledge sources to constrain speech systems. The transition probabilities of the sequence of sound units is by default used in n-gram language modelling to serve as constraints to the speech recognizer, and is derived automatically by standard language modelling toolkit [148]. In [149] transition probabilities between linguistic units has been used to evaluate the performance of speech recognition systems.

Table 3.9 Transition probabilities between consonant groups in Tamil.

	Freq.	$< /s >$	Velar	Retroflex	Dental	Bilabial	Nasal	Vowel	Semivowel	Fricative	Affricate	Flap	Lateral	Trill
Velar	13244	0.183	0.2652	0.0065	0.0010	0.0003	0.0005	0.7038	0.0005	0.0025	-	0.0053	0.0042	0.0002
Retroflex	9845	0.009	0.0063	0.2244	0.0003	0.0095	0.0006	0.7199	0.0006	0.0001	0.0329	0.0049	0.0004	0.0001
Dental	17562	0.089	0.0002	-	0.1599	-	0.0004	0.8321	0.0018	0.0003	0.0001	0.0049	0.0001	0.0001
Bilabial	9602	0.009	0.0005	0.0012	0.0012	0.2031	0.0001	0.7546	0.0010	0.0001	-	0.0334	0.0008	-
Nasal	24064	0.194	0.0551	0.0504	0.1024	0.0232	0.0429	0.6393	0.0201	0.0048	0.0129	0.0013	0.0003	0.0743
Vowel	106320	0.165	0.0453	0.0707	0.0904	0.0316	0.1987	0.0632	0.1105	0.0631	0.0117	0.1315	0.1355	0.0478
Semivowel	13096	0.008	0.0008	-	0.0229	0.0018	0.0018	0.9575	0.0127	0.0005	0.0011	0.0003	0.0002	-
Fricative	8529	0.044	0.0022	0.0055	0.0438	0.0020	0.0189	0.9043	0.0121	0.0013	-	0.0038	0.0041	0.0018
Affricate	3310	0.448	0.0003	-	-	0.0164	0.0006	0.7512	0.0006	0.0013	0.2289	0.0003	0.0003	-
Flaps	1296	0.237	0.0325	0.0007	0.0295	0.0169	0.0209	0.7939	0.0104	0.0905	0.0037	-	0.0004	0.0005
Lateral	13378	0.304	0.0170	0.0039	0.0067	0.0042	0.0153	0.7888	0.0251	0.0104	0.0110	0.0001	0.1175	0.0001
Trill	6741	0.002	0.0746	-	0.0003	0.0213	0.0004	0.7416	0.0001	-	0.0076	0.0010	-	0.1532

Table 3.10 Transition probabilities between consonant groups in Telugu.

	Freq.	$< /s >$	Velar	Retroflex	Dental	Bilabial	Nasal	Vowel	Semivowel	Fricative	Affricate	Flap	Lateral
Velar	10751	0.021	0.0160	0.0215	0.0093	0.0017	0.0025	0.8153	0.0219	0.0445	0.0009	0.0620	0.0035
Retroflex	6117	0.050	0.0009	0.0587	0.0003	0.0005	0.0091	0.8074	0.0057	0.0010	0.0009	0.0532	0.0625
Dental	11858	0.020	0.0010	-	0.0412	0.0011	0.0039	0.8060	0.0558	0.0055	-	0.0817	0.0037
Bilabial	8804	0.009	0.0009	0.0036	0.0095	0.0478	0.0015	0.7510	0.0179	0.0010	0.0002	0.1621	0.0044
Nasal	22766	0.088	0.0490	0.0391	0.1110	0.0205	0.0740	0.6020	0.0080	0.0128	0.0522	0.0026	0.0285
Vowel	84351	0.248	0.1051	0.0512	0.0968	0.0525	0.2802	0.0007	0.0705	0.0769	0.0255	0.1348	0.1058
Semivowel	8852	0.015	0.0005	0.0003	0.0002	0.0002	0.0005	0.9590	0.0317	0.0009	0.0001	0.0064	0.0001
Fricative	9693	0.052	0.0100	0.0582	0.0913	0.0098	0.0316	0.6750	0.0218	0.0094	0.0033	0.0853	0.0042
Affricate	5004	0.014	0.00060	0.0002	0.0002	0.0111	0.0016	0.9426	0.0083	0.0010	0.0318	0.0010	0.0014
Flaps	14044	0.041	0.0226	0.0272	0.0236	0.0135	0.0198	0.8106	0.0357	0.0168	0.0104	0.0007	0.0192
Lateral	8969	0.039	0.0110	0.0021	0.0031	0.0086	0.0079	0.8814	0.0057	0.0140	0.0022	0.0007	0.0633

Table 3.11 Transition probabilities between consonant groups in Hindi.

	Freq.	$< /s >$	Velar	Retroflex	Dental	Bilabial	Nasal	Vowel	Semivowel	Fricative	Affricate	Flap	Lateral
Velar	10275	0.0879	0.0016	0.0070	0.0204	-	0.0007	0.9052	0.0139	0.0275	-	0.0173	0.0063
Retroflex	2088	0.2055	0.0018	0.0096	0.0030	0.0012	0.0108	0.8813	0.0036	0.0024	-	0.0856	0.0006
Dental	6465	0.2196	0.0004	-	0.0314	0.0004	0.0031	0.8323	0.0360	0.0021	0.0006	0.1028	0.0002
Bilabial	5542	0.0655	-	0.0033	0.0133	0.0030	0.0012	0.8757	0.0027	0.0025	0.0041	0.0909	0.0037
Nasal	12167	0.2752	0.0427	0.0262	0.0630	0.0146	0.0178	0.7738	0.0196	0.0272	0.0120	0.0018	0.0011
Vowel	54768	0.2734	0.0904	0.0300	0.0843	0.0424	0.2147	0.1301	0.0785	0.1067	0.0344	0.1345	0.0538
Semivowel	5082	0.0551	0.0010	-	0.0002	0.0002	0.0002	0.9767	0.0181	0.0021	-	0.0015	-
Fricative	9457	0.1242	0.0053	0.0240	0.0625	0.0054	0.0089	0.8358	0.0144	0.0059	0.0037	0.0304	0.0035
Affricate	2818	0.1600	-	-	0.0004	0.0021	0.0008	0.9480	0.0232	0.0004	0.0215	-	0.0034
Flaps	7746	0.2894	0.0093	0.0200	0.0213	0.0071	0.0238	0.8699	0.0199	0.0158	0.0087	0.0035	0.0007
Lateral	3084	0.2231	0.0013	0.0038	0.0054	0.0046	0.0042	0.0998	0.0050	0.0100	0.0008	-	0.0651

3.4 INFORMATION THEORETIC PERSPECTIVE ON SUBWORD UNITS FOR SPEECH RECOGNITION

Most speech recognition systems use phonemes and context dependent biphones or triphones as the subword unit to model the acoustics of the speech signal. The difficulty with the use of phonemes is that they are often deleted/merged in continuous speech. The advantages and disadvantages of the use of these units is detailed in Section 4.1. Modelling speech as a sequence of phonemes is not entirely valid, as a one-to-one mapping between the acoustic signal, and the phoneme sequence supposed to have been spoken is not straightforward. Using biphones and triphones as subword units may improve the performance of the recognizer, as some amount of local constraints are incorporated. As more amount of local constraints are incorporated, the performance of the speech recognizer improves. In this section we show, from an information theoretic perspective, that syllable is a natural linguistic unit that captures the constraints in the language better, and hence may be an appropriate unit for speech recognition.

3.4.1 Tamil Text corpus

The statistical properties of various subword units are studied in the information theoretic perspective. This is illustrated using the transcribed text of the Tamil speech database. In Tamil there are 35 phonemes native to the language [150]. In addition, to handle borrowed words from other languages, especially Sanskrit, this set is augmented with 5 phonemes. Altogether for this study we considered a set of 40 phonemes for the language, of which 38 phonemes are present in the database. The database contains 1975 distinct syllables. The analysis is done for different subword units like phonemes, biphones, triphones and syllables.

3.4.2 Entropy and redundancy computation

Consider the natural language spoken as one generated from a source (S) containing q symbols, and that the source symbol s_i is dependent on a finite number m of preceding symbols. Such a source is called an m^{th} order Markov source [151]. For an m^{th} order Markov source the probability of emitting a given symbol is known if we know the preceding m symbols. At any given time the m preceding symbols are called the state of the m^{th} order Markov source at that time. As there are q possible symbols, an m^{th} order Markov source will have q^m possible states.

If we are in a state specified by $(s_{j_1}, s_{j_2}, \cdots, s_{j_m})$, i.e., the previous m symbols emitted are $s_{j_1}, s_{j_2}, \cdots, s_{j_m}$, then the conditional probability of receiving symbol s_i is $P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m})$. Following [151], the average amount of information per symbol while in state $(s_{j_1}, s_{j_2}, \cdots, s_{j_m})$ is given by

$$H(S|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) = -\sum_{S} P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) \log P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) \quad (3.1)$$

Averaging the quantity over all q^m possible states, the entropy of the m^{th}-order Markov source S is given by

$$H(S) = -\sum_{S^m} P(s_{j_1}, s_{j_2}, \cdots, s_{j_m}) \sum_{S} P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) \log P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) \quad (3.2)$$

If S is a zero-memory source rather than Markov, then $P(s_i|s_{j_1}, s_{j_2}, \cdots, s_{j_m}) = P(s_i)$ and (3.2) reduces to

$$H(S) = -\sum_{S} P(s_i) \log P(s_i) \quad (3.3)$$

The relative entropy is given by

$$H_{rel} = \frac{H(S)}{H_{max}} \quad . \quad (3.4)$$

The H_{max} occurs when all the q symbols are equiprobable, and it is given by $\log_2 q$. Redundancy (R) measures the amount of constraints imposed on the text in the language due to its syntactical rules. Every syntactic rule in the language imposes some constraints, which in turn introduces some redundancy [152]. The redundancy is defined as the difference between H_{max} and $H(S)$, expressed as a fraction of H_{max}.

$$R = \frac{H_{max} - H(S)}{H_{max}} = 1 - H_{rel} \quad . \tag{3.5}$$

Just as entropy measures on an average, the uncertainty of the outcome of an event, we can view redundancy as an average measure of our confidence in the outcome. It is difficult to correctly estimate the number of words or syllables (units) in a language. In most speech recognition or NLP tasks it is not required to do so. The development of a lexicon is task specific and is also governed by the training data set. For a specified training set and lexicon the number of triphones or syllables are fixed. Even in such situations the number of triphones exceeds the number of syllables. The large sized units capture the coarticulation among the phonemes and are better identified in speech. Among the multiphone models, triphones and cross word triphones are popularly used. These units led to a progressively better performance of the speech recognizer. Such speech units are not naturally observed in speech. These units are forcefully created from a machine recognition point of view. The main purpose of creating larger sized units is to incorporate constraints within the unit. When more constraints are present in the unit it will be recognised better in a speech recognizer. Syllable is a natural unit and is easily identified in the speech signal. The syllables capture the coarticulation among the sound units efficiently and naturally. Since syllables are tightly bound by the rules of the language they have more constraints incorporated within them (compared to phonemes/biphones). For such units the redundancy will be

higher. Compared to the triphones this higher redundancy of the syllables is possible due to the smaller symbol set. The smaller symbol set for the syllable will also reduce the confusability and improve the recognition rate of the syllables. Hence it desirable to use syllable as a unit for speech recognition.

The efficiency of five different subword units used in speech recognition, namely phonemes, context dependent biphones and triphones, nonoverlapping triphones and syllables in capturing the syntactic constraints is analysed. Pronunciations of the words in the database in terms of phonemes are determined. These phonemes are the first set of subword units considered. Using these phonemic representation of the words, within word left context dependent biphones and context dependent triphones are determined in a conventional manner as used in the standard speech recognition toolkits [153]. For example, the word [*pala*] (meaning many) would be parsed into phonemes as /p a l a/, into biphones and monophones as /p-a a-l l-a/, and into triphones and biphones as /p+a p-a+l a-l+a l-a/. Similarly it would be parsed into nonoverlapping triphones, biphones and monophones as /p-a+l a/ and into syllables as /pa la/. The context dependent biphones along with the monophones at word beginnings are 807 in number, out of the possible $38^2 + 38 = 1482$. These symbols form the second set of subword units for analysis. The triphones and their associated word-begin and word-end biphones that occurr in this word set are 5,286 of the possible $38^3 + 38^2 = 56316$ units. These symbols are the third set of subword units used. The syllables are derived by parsing the text based on rules of the language. A total of 1975 distinct syllables occur in the database. The structure of the biphones and triphones are driven from a machine recognition point of view. The structure of a triphone is such that the center phoneme of one triphone becomes the left phoneme of the next triphone. Such overlaps are not seen in the syllables, with the exception of consonant

geminates (e.g. *vanakkam /va nak kam/*). To compare the properties of the syllable with a structurally similar unit, the nonoverlapping triphone is generated. This unit is obtained by concatenating three phonemes of the same word. These nonoverlapping triphones and the associated word-end biphones are similar to syllables in construction but nonlinguistic in nature. In the context of speech recognition, to analyse how well each unit individually captures the constraints, we assume a zero memory source approximation of the language. For the 5 different units, the entropy and redundancy are shown in Table 3.12.

Table 3.12 Entropy and redundancy of the subword units for a zero-memory source approximation.

Subword unit	Size of symbol set	H_{max} (bits/symbol)	Entropy (bits/symbol)	Redundancy (%)
Phoneme	38	5.25	4.55	13.33
Biphone	807	9.66	7.82	19.05
Triphone	5286	12.37	10.16	17.87
Nonoverlapping triphone	3164	11.62	9.43	18.85
Syllable	1975	10.95	8.43	23.01

The characteristics of the five subword units in approximating the language as a first and second order Markov source is studied. The entropy and redundancy of the subword units is given in Tables 3.13 and 3.14. For the first and second order Markov source assumption, the redundancy for the syllable is better than that of the phoneme and biphones. In the case of second order Markov source assumption, the redundancy of the syllables is comparable to that of triphones, but achieved with a smaller set of

Table 3.13 Entropy and redundancy of the subword units for a first order Markov source approximation.

Subword unit	Size of symbol set	H_{max} (bits/symbol)	Entropy (bits/symbol)	Redundancy (%)
Phoneme	38	5.25	2.92	44.38
Biphone	807	9.66	2.12	78.05
Triphone	5286	12.37	1.30	89.49
Nonoverlapping triphone	3164	11.62	1.52	86.92
Syllables	1975	16.95	2.33	78.72

units. This is useful for reducing the search space in speech recognition. As mentioned earlier, due to large redundancy built into triphones by way of construction, and due to the additional constraints of a first and second order Markov source, we find that the redundancies of the triphones for a first order Markov source to be higher than the redundancies of the syllables. Studying the redundancies, we hypothesize that for recognition of subword units in speech, syllable as a unit may perform better than the other units considered in this study. Due to the high first and second order redundancies of the syllable, a syllable-based n-gram language model is likely to further improve the performance of the speech recognizer.

3.5 SUMMARY

In this chapter, we studied the statistical characteristics of the basic subword units and the words in the language. The suitability of a subword unit for speech recognition using information theoretic principles was studied. A unit that incorporates a large

Table 3.14 Entropy and redundancy of the subword units for a second order Markov source approximation.

Subword unit	Size of symbol set	H_{max} (bits/symbol)	Entropy (bits/symbol)	Redundancy (%)
Phoneme	38	5.25	2.04	61.14
Biphone	807	9.66	1.35	86.02
Triphone	5,286	12.37	0.73	94.10
Nonoverlapping triphone	3,164	11.62	0.51	95.34
Syllables	1,975	10.95	0.74	93.24

amount of constraints within itself would perform better for the speech recognition task. For such units, the redundancy would be higher. We find that the syllable has a higher redundancy compared to the other conventionally used subword units like phonemes, biphones and triphones. The frequency of occurrence of the syllables in the corpus was studied. This study revealed that a small subset comprising of the most frequently occurring syllables cover nearly 89% of the occurrence of the syllables in the database. Thus, it is essential to focus on modelling of these syllables. The frequency of occurrence of the different word and syllable patterns (e.g, CV, CCVC, CVCC) were studied. A study of the duration of the syllables in continuous speech revealed that the most frequently used syllables are of smaller durations, while less frequently used syllables, on the average, have larger durations. The knowledge of the duration and the frequency of occurrence of syllables was found useful in the development of the HMM models. In the next chapter, we discuss the development of a syllable recognizer using the framework of HMM. This is illustrated for the two Indian languages, Tamil

and Telugu. Approaches to model the dynamics of a syllable, and methods to reduce the search space in the recognition system are discussed.

Table 3.15 Summary of studies on the statistical characteristics of sound units in Indian languages.

- Knowledge of the characteristics and distribution of the sound units in a language is essential for the development of a speech recognizer.
- The frequency of occurrences of the different sound units in a language were studied. These studies were useful in deciding on the parameters of the HMM models in the recogniser.
- From an information theoretic perspective it was shown that syllables seem to be an appropriate unit for speech recognition.

CHAPTER 4

RECOGNITION OF SYLLABLES IN CONTINUOUS SPEECH

4.1 INTRODUCTION

In this chapter the focus is to develop a speech recognizer capable of recognizing the sequence of the basic subword units like syllables with the highest accuracy possible. Once the basic subword units in the speech signal are recognized, it is easier to use a lexicon of words in terms of syllables to identify the word corresponding to the recognized syllable sequence. In the following sections the development of a HMM-based syllable recognizer for continuous speech in Tamil and Telugu is described.

The number of distinct syllables that occur in a language is large. A syllable recognizer with a vocabulary encompassing all these syllables would perform poorly. This is due to the insufficient examples of all the syllables required to train the models, and due to the large search space. The frequency of occurrence of the syllables in the database is given in Table 4.1. It is observed that about half of the syllables in the database have less than 5 examples. To train a HMM model we assume a minimum of at least 50 training examples to be present in the database, to consider the syllable as part of the vocabulary. Thus a set of 328 syllables for Tamil and 267 syllables for Telugu formed the syllable vocabulary. The majority of these syllables are of CV or CVC type. A syllable recognizer is developed for this vocabulary.

4.2 PRELIMINARY SPEECH RECOGNITION SYSTEM

An unconstrained syllable recognizer in which any syllable in the vocabulary follows any other syllable can be modelled as shown in Fig. 4.1 for the two languages. No

Table 4.1 Varying # occurrences of the syllables in the database.

# occurrence	# Syllables	
	Tamil	Telugu
Greater than 1000	13	10
501 to 1000	32	25
101 to 500	148	132
5 to 100	817	512
Less than 5	938	1294

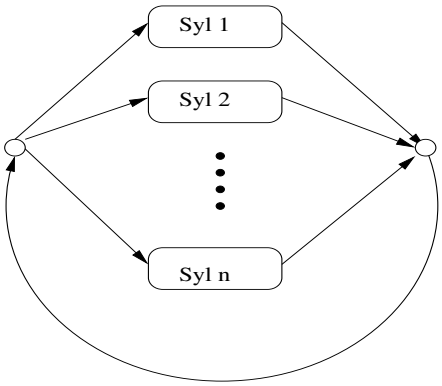

Fig. 4.1 Syllable loop grammar used for recognition.

silence model is used. The lexicon for this task maps a syllable onto itself.

The speech corpus is split into training and test sets as given in Table 4.2. The database contains information about the manually marked syllable boundaries. Using this information from the continuous speech utterance in the training set, segments corresponding to each syllable in the utterance are spliced out. All the segments of speech corresponding to a syllable are pooled together for training the syllable. From each speech frame of 15 msec, shifted by 5 msec, a 39 dimension vector, comprising of 13

Table 4.2 Description of Tamil and Telugu language data sets used for speech recognition.

Language		# Bulletins		# speech segments	Total # words	# distinct words	Total # syllables	# distinct syllables
		Male	Female					
Tamil	Train	08	21	6410	26903		88169	
	Test	02	02	949	3785		12538	
	Total	10	23	7359	30688	8947	100707	1975
Telugu	Train	09	08	5501	21400		71005	
	Test	02	01	983	4063		13344	
	Total	11	09	6484	25463	9218	84349	2273

dimension mel-frequency cepstral coefficients, along with their delta and acceleration coefficients, is derived [153]. The acoustic model for each syllable is a 5-state left-to-right HMM with entry and exit states [11]. The models are trained in an isolated word fashion. This forms the preliminary system. This system is tested against a set of 949 continuous speech utterance in Tamil and 983 utterances in Telugu, which form the test set as given in Table 4.2. The output of the syllable recognizer for a continuous speech test utterance is the sequence of syllables hypothesised for that utterance. As the syllable vocabulary is a subset of the syllables in the database, many Out of Vocabulary (OOV) syllables that appear in the test utterance would be hypothesised with the closest syllable in the vocabulary. This results in substitution errors proportional to the OOV rate. The performance of the recognizer is given in terms of the number of syllables recognized correctly out of the total number of valid syllables (syllables in the vocabulary) that occur in the test utterance (%correct). The

Table 4.3 Performance of preliminary syllable recognition system.

Language	Syllable recognition (% correct)
Tamil	24.2
Telugu	26.6

Table 4.4 Details of the number of mixtures used for syllables with varying frequency of occurrence for Tamil and Telugu syllable recognizers.

# examples	# mixtures used	# syllables	
		Tamil	Telugu
> 1000	64	13	10
701 to 1000	32	20	17
401 to 700	16	25	18
201 to 400	8	58	48
101 to 200	4	77	71
≥50	2	135	103

performance of the preliminary system is given in Table 4.3.

4.3 IMPROVEMENTS TO THE PRELIMINARY SYSTEM

The preliminary system uses a single Gaussian per state of the HMM model. As the distribution of the feature vectors can be better modelled using a mixture of Gaussians, based on an empirical guideline of using one mixture per 25 examples [32], we chose the number of mixtures per state per model as given in Table 4.4. The performance

Table 4.5 Recognition performance using syllable level bigram language model.

Language	Syllable recognition (% correct)	Relative improvement (%)
Tamil	57.6	36
Telugu	61.0	19

of this system with a mixture of Gaussians improves the recognition rate to 42.4% for Tamil and 51.2% for Telugu. The local syntactic constraints of a language can be incorporated using language models. The language models are sensitive to the type and genre of data used. Due to the lack of large text corpus of the same genre as the TV news data, the transcription of the limited training corpus is used to derive a syllable level bigram language model. The local constraints at the syllable level when incorporated as a bigram language model in the syllable recognizer, improves its performance as shown in Table 4.5. The relative improvement over the Gaussian mixture system is 36% for Tamil and 19% for Telugu.

4.4 MODELLING THE DYNAMICS OF THE SYLLABLES

Syllables are dynamic sounds. Some of the regions like closure, burst, aspiration, transition and steady vowel regions are observed in their acoustic signals. The states of a HMM corresponds to the events in the speech signal. For some syllables the change in the dynamics of the acoustic signal may be such that it may be inefficient to characterise them by a 5-state HMM. Conversely, it is not desirable to use a large number of states for all the syllables, as the parameters for all the models may not be estimated accurately due to limited amount of training data. As a trade off, we use

Table 4.6 The size of the syllable set and the number of states used for modelling the HMMs in Telugu.

# states in the model	# syllables modelled
4	17
5	77
6	91
7	48
8	27
9	7

half the number of frames in the median duration of the syllables as the basis for the number of states to be used [74]. To study the effect of modelling the dynamics of the syllable using larger number of states, experiments are conducted using Telugu only, as an illustration. The language model and Gaussian mixtures are not used for the study.

The 267 syllables in the vocabulary of Telugu are modelled using 4 to 9 states, depending on the durations of the syllables. The number of states used for the syllables is varied as given in Table 4.6. The recognition rate of the syllables for the continuous speech test utterances is determined. We observe that 58% of the syllables perform better with increased number of states than their 5 state counter parts. For 17% of the syllables there is no improvement due to the change in number of states derived based on their median duration. In 25% of the cases where syllables have their number of states larger than 5, the performance decreased marginally.

To study the effect of increase in the number of states in a model, the recognition

rates of individual syllables is determined. Based on the number of states in the median duration of the syllable, a set of 17 syllables are modelled with 4 states as given in Table 4.6. Among these syllables, 13 performed poorly as compared to the 5-state models. This suggests that it is inappropriate to model the dynamic syllabic units with 4 states or less. When the syllables are modelled with larger number of states (6 and above) the performance of the syllable recognizer improves. The recognition rate of some of the syllables modeled with 5 and 6 state HMMs is shown in Table 4.7. All the syllables shown here have improved recognition rates.

Table 4.7 Recognition performance for some CV units in Telugu for increased number of states.

Syllable	Recognition performance % correct		
	4-State	5-State	6-State
/Da/	18.4	23.7	
/Du/	24.7	40.2	
/la/	07.3	12.5	
/lu/	44.4	49.8	
/ru/	33.3	46.3	
/Ti/		32.8	50.0
/Tu/		26.9	30.8
/ja/		07.4	15.8
/ka/		04.3	10.2
/na/		18.9	21.8
/pa/		12.3	25.9

For a continuous speech test utterance the optimal path chosen by the decoder is influenced by the number of states in each model. If the number of states in a model change, then the optimal path taken while decoding could be different, affecting the overall performance for continuous speech. To study the significance of the increase in the number of states, an arbitrary set of 34 CV and CVC type of syllables are chosen, and the number of states for each of the syllables are varied from 4 to 10. The models are trained and tested in an isolated word fashion. Gaussian mixtures are not used to represent the distribution of feature vectors. The recognition rate of the syllables for the 5-state model and for the model with the optimal number of states is given in Table 4.8. It is observed from the Table 4.8 that for some syllables the 5-state model performs best. For all other syllables, the models with larger number of states perform better. The improvement in performance of these models over that of the 5-state model varies from 1% to 55%. This suggests that we may be able to model the dynamics of the syllables better with larger number of states in the HMM framework. Using varying number of states for the syllables in the syllable recognizer, of Section 4.3, we observe that the performance of the syllable recognizer improves by 4% in relative terms as given in Table 4.9. This is the best syllable recognition rate achieved for this database.

Table 4.8 Recognition rate of a subset of 37 syllables in Telugu for a 5-state model and a model with optimal number of states.

Syllable	Optimal # states	5-state model (%) correct	Optimal # states (%) correct	Syllable	Optimal # states	5-state model (%) correct	Optimal # states (%) correct
/Da/	8	13.2	40.5	/vE/	7	49.0	55.1
/Du/	6	58.1	58.1	/pA/	5	70.4	70.4
/bu/	6	91.7	100.0	/pra/	9	49.6	60.3
/da/	5	52.4	52.4	/man/	8	87.5	94.4
/du/	6	33.3	37.5	/kAr/	7	32.7	76.9
/gi/	4	55.6	63.0	/pAr/	9	22.6	77.4
/gu/	8	49.1	58.2	/tri/	5	82.1	82.1
/la/	8	55.1	62.0	/tun/	6	78.0	82.9
/li/	8	64.0	64.9	/ram/	6	60.0	60.0
/ra/	8	19.5	25.2	/sam/	6	90.2	98.0
/ri/	9	49.5	61.5	/rASh/	7	98.1	100.0
/ru/	8	56.4	62.0	/mukh/	10	40.6	75.0
/va/	7	23.5	27.8	/chin/	-	100.0	100.0
/vu/	5	70.0	70.0	/bAd/	8	57.9	68.4
/ya/	7	24.5	33.2	/gam/	8	44.4	55.6
/yi/	8	75.9	71.3	/mup/	6	75.0	83.3
/yu/	6	19.4	29.0	/Tram/	-	100.0	100.0
/DA/	9	36.6	63.4	/hA/	8	65.8	81.6

Table 4.9 Recognition performance using larger number of states.

Language	Syllable recognition (%)	Relative improvement (%)
Tamil	59.9	4
Telugu	63.5	4

4.5 EQUIVALENT CLASSES FOR RECOGNITION OF SYLLABLES

In certain application like man-machine dialogue and command and control applications, keywords in the spoken utterances are of interest. As the task is generally of limited vocabulary and restricted domain, it is preferable to map all the pronunciation variations of similar syllables to one base form. It may also be desirable not to distinguish between similar consonants like /D/ and /d/. For example, the pronunciation variation due to dialect and accents or mispronunciation like /maD-rAs/ and/mAd-ras/ may all have to be mapped to the base form of the word /mad-ras/ in a dialogue system. The simplest approach for such an application is that, equivalent classes of syllables are formed by combining syllables that have the same consonant clusters, but a long or short version of the vowel ({/ka/,/kA/} ->/KA/). Syllables with consonants that sound similar are mapped onto one equivalent class ({/na/,/NA/,/Na/} -> /NA/). In the syllable recognizer of Section 4.4, if such a mapping to equivalent classes is carried out, the number of distinct syllables (classes) in the vocabulary reduces from 267 to 220. Using the equivalent class label as the output of the syllable recognizer, the performance of the recognizer improves to 62.1% for Tamil and 67.5% for Telugu.

4.6 EQUIVALENT CLASS BASED PRECLASSIFICATION FOR RECOGNITION OF SYLLABLES

Hidden Markov models are capable of modelling the variability in the speech sound, and absorbing the variability in the duration of a sound unit. Discriminative training is not normally provided in these models. Neural networks on the other hand provide discriminative training, and are good as classifiers. The neural networks are capable of capturing the nonlinear surfaces separating different classes. But they require fixed dimension patterns representing the speech sound. It is possible to combine the strengths of HMM and neural network to improve the recognition rate.

In this section we explore a method of enhancing the recognition rate of syllables extracted from continuous speech by preclassifying them using a neural network based preprocessor. This preclassification step is followed by recognition of the syllables using conventional HMM models. The syllables are initially grouped into a set of equivalent classes based on their syllable structure. The syllables within an equivalent class are called 'cohorts'. A support vector machine model is trained to classify the syllable into the equivalent class to which it belongs. In the next stage, a HMM-based syllable recognizer is used to recognize the syllables within the equivalent class. This reduces the search space of the recognizer from the list of syllables in the vocabulary to the number of syllables within a particular equivalent class. This approach reduces the confusability among the syllables and increases the recognition rate. The feasibility of this approach is studied for the Telugu language, as an illustration.

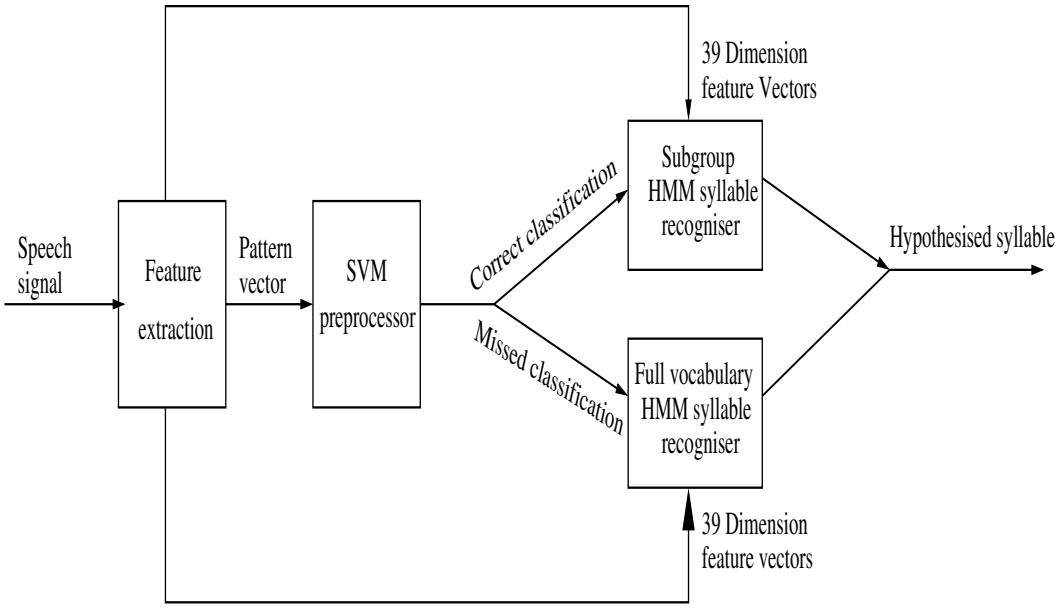

Fig. 4.2 Block diagram of SVM-HMM hybrid system.

4.6.1 System description

The block diagram of the proposed system is shown in Fig. 4.2. The syllable boundary information available in the database is used to segment the speech signal into syllables. For every syllable segment, 13 dimension MFCC are extracted for every frame of 15 msec with a frame shift of 5 msec [11]. A fixed 130 dimension pattern vector, is formed by concatenating the 10 frames corresponding to the syllable segment. These frames are chosen by the method of linear compaction and elongation as described in [154]. These large dimension pattern vectors are expected to capture the dynamics of the syllables. These pattern vectors are used to train a 6 class SVM classifier. The equivalent class to which a syllable maps to is determined by replacing all the consonants in it with 'C' and the vowels by 'V'. The resulting syllable structure represents the equivalent class label for the syllable. The 267 syllables in the Telugu vocabulary map onto 6 equivalent classes. The patterns corresponding to all the cohorts in an

equivalent class forms the training set for that class. The SVM classifier is trained to map the input pattern into one of these six classes. In the testing stage, the output of the preclassifier is the equivalent class label of the syllable being recognized. If the group classification is correct, then a HMM-based syllable recognizer that recognizes the cohorts within that class is used. For a missed classification, an alternative HMM-based syllable recognizer developed for the entire syllable vocabulary is used for recognition. The prior knowledge of the equivalent class is used to hypothesize the correct/missed classification.

4.6.2 Database used in the study

The objective is to study if a preclassifier can improve the performance of the syllable recognizer. This study is illustrated using the Telugu language database. The syllables in the vocabulary form the following equivalent classes: CCV, CCVC, CV, CVC, V and VC. The training and test dataset of syllables spliced from continuous speech for the equivalent classes is shown in Table 4.10.

4.6.3 SVM preprocessor

The SVM preclassifiers are trained with the 130 dimension pattern vectors. One against the rest approach is used for this multiclass task. Given a test pattern vector, the class label of the SVM model which gives the highest score is hypothesized as the class label for the test pattern. The performance of the SVM preclassifier is given in Table 4.11. It is observed that the recognition rate of frequently occurring classes, namely, CV, CVC and V, is high. The overall classification rate is 87.17%.

4.6.4 HMM-based syllable recognizer

For the SVM-HMM hybrid system, in the case of a correct classification by the SVM preclassifier, a HMM subgroup recognizer is required. For a misclassification by the

Table 4.10 Syllables in the vocabulary categorised into equivalent classes for the Telugu language database.

Equivalent class	# training examples	# testing examples
CCV	2660	427
CCVC	248	41
CV	44499	8333
CVC	9361	1774
V	2348	472
VC	1031	185
Total	59547	11232

Table 4.11 Performance of the SVM preprocessor in recognizing equivalent classes.

Equivalent class	Recognition rate (% correct)
CCV	53.40
CCVC	48.78
CV	94.64
CVC	69.22
V	70.76
VC	52.43
Overall performance	87.17

SVM preclassifier, a whole vocabulary syllable recognizer is required. There are 6 equivalent classes. A HMM-based subgroup recognizer is developed to recognize the cohorts within each of the 6 equivalent classes. The full vocabulary recognizer recognizes one among the 267 syllables in the vocabulary. The syllable loop grammar is used for all the recognizers. All recognizers use 8-state HMM models and 2 to 64 Gaussian mixtures per state depending on the training examples available for each syllable. The models are trained and tested in an isolated word fashion. The performance of the full vocabulary recognizer, without the use of language models, is 56.85%. The performance of the recognizer in Section 4.3 for continuous speech was 52.73%. The results are given in Table 4.12.

Table 4.12 Comparison of performance of the preprocessor based hybrid system and the HMM-based system for recognition of syllable segments in continuous speech.

Recognition system	Percentage of correct classification
HMM-based continuous speech recognition system without syllable boundary information	52.73
HMM-based speech recognition system with syllable boundary information	56.85
SVM-HMM hybrid recognizer with syllable boundary information	61.58

4.6.5 SVM-HMM Hybrid System

The results of the preprocessing stage is the class label of the syllable. If the output of the SVM preprocessor is correct, then the subgroup HMM syllable recognizer appro-

priate for that equivalent class is used to recognize the syllable. On the other hand, for a missed preprocessor classification the global HMM syllable recognizer designed for the entire syllable vocabulary is used for recognition. The performance of the hybrid SVM-HMM system is given in Table 4.12. We observe that the performance of the hybrid system is better than the HMM-based syllable recognizer.

Conventionally the knowledge of the equivalent class label is not available at the testing stage. The recognition system would be treated as a black box. Given an input signal, the syllable label is the output desired. In such a situation the class label output by the SVM preprocessor is taken as the correct classification. The subgroup recognizer within this class is used to recognize the syllable in the speech segment. The performance of the SVM-HMM system is 54.44%. When two systems are cascaded together then the performance of the combine system is influenced by the first system. For the SVM-HMM preprocessor system the expected performance is $87.17 \times 56.85 = 49.55\%$. The actual performance the hybrid system is better than this.

In this section we showed that preprocessing followed by recognition can reduce the confusion and search space among the syllables. This improves the recognition rate of the syllables. Here knowledge of the class labels is used in testing. The performance of the hybrid system when treated as a black box is reduced due to the limited performance of the individual systems. The performance of the syllable recognizer may further be improved by using the n-best output of the hybrid system and rescoring the hypothesis in using a syllable bigram language model.

4.7 SUMMARY

In this chapter, we developed a syllable recognizer capable of recognizing the syllables in continuous speech. This preliminary system used standard 5-state left-to-right

HMM models. The performance of this system was improved by using a mixture of Gaussians and bigram level language models. To improve the recognition rate of the syllables, SVM-based preclassifiers was proposed to reduce the search space. This approach improved the recognition rate of the syllables extracted from continuous speech, and tested in an isolated word fashion. The knowledge of the group to which the syllable belongs was used. Incorporating language models into this recognizer may improve its performance further. In the next chapter we discuss the basic concepts of LSA and the development of a word based large span constraint language model. We also extend this concept to show that latent large span constraints exist at the syllable level also.

Table 4.13 Summary of the studies in development of the syllable recognizer for Indian languages.

- Syllables are ideal subword units for speech recognition. They are highly confusable units and their dynamics change rapidly. These units need to be modelled accurately.
- Previous approaches have used HMM and ANN models for the task.
- In this work we developed a syllable recogniser for Indian languages. The dynamics of the syllables were modelled using larger number of states.
- A SVM based preclassifier was proposed to reduce the search space in the recogniser.

CHAPTER 5

LSA FOR LANGUAGE MODELING

5.1 INTRODUCTION

Constraints exist at various levels in natural languages. As mentioned in Chapter 1, the constraints exist from the phonetic level to the pragmatic level. Modelling these constraints is important for most speech systems. These constraints can be captured using language modelling techniques. The models help determine the permissible sequences of words and sounds in a language. The most popular language models are the n-gram models. The ability of the n-gram models to capture the constraints depends upon its ability to distinguish between different strings of n-words. When the value of n is large, the parameter estimates of the n-gram model is poor. Consequently, the predictive power of the n-gram model is reduced.

Semantic constraints that are spread over large spans need to be captured. One of the approaches in modelling these large span constraints is LSA. The LSA uses the information about the co-occurrence of words in the entire document to derive the relationship among the words. Since the scope of the analysis is the entire document, the approach is able to capture the semantic constraints over large spans. Latent semantic analysis approach originally designed for information retrieval can be adapted to language modelling. The LSA paradigm can be integrated with n-gram language models. It is shown that this integration reduces the perplexity to a substantial extent. The development of the LSA language model and its integration with the n-gram language model as detailed in [7] and [8] is described here. The approach presumes the availability of a database of documents. The documents should preferably contain

a set of sentences that are semantically homogeneous, describing a concept or a story.

5.1.1 Feature extraction

Let ν, $\mid \nu \mid = M$, be some vocabulary and $\mathcal{T}$ a training text corpus, comprising of N articles (documents) relevant to the domain of interest. Typically M and N are of the order 10000 and 100000, respectively. $\mathcal{T}$ might comprise of a hundred million words or so.

The LSA paradigm defines a mapping between the discrete sets ν and $\mathcal{T}$ and a continuous vector space Ψ, whereby each word w_i in ν is represented by a vector $\bar{u}_i$ in Ψ, and each document d_j in $\mathcal{T}$ is represented by $\bar{v}_j$ in Ψ.

The first in the development of a LSA-based model is the construction of a matrix $\mathbf{W}$ of co-occurrences between words and documents. Here word order is ignored unlike n-gram modelling. LSA is a bag of words approach that disregards co-locational information in word strings. The context of a word is in that sense the entire document. Which word is associated with what document is the information sought to be captured. The word count, i.e., the number of times a particular word appears in a particular document, is used. The documents will be of different lengths. Based on experiments in the field of information retrieval, it is preferable to normalise the word count for the document length and the word entropy. Thus every cell (i, j) in $\mathbf{W}$ is given by

$$w_{i,j} = (1 - \epsilon_i)\frac{c_{i,j}}{n_j} \tag{5.1}$$

where,

$c_{i,j}$ is number of times word w_i occurs in document d_j,

n_j is total number of words present in d_j,

ϵ_i is normalised entropy of w_i in the corpus $\mathcal{T}$

Weighting the word count by $(1 - \epsilon_i)$ means that two words that appear the

same number of times in a document do not necessarily convey the same amount of information about the document. This depends on the distribution of the words in the document collection $\mathcal{T}$. The value for ϵ_i is obtained as follows:

If we denote $t_i = \sum_j c_{i,j}$ as the total number of times w_i occurs in $\mathcal{T}$, then

$$\epsilon_i \quad = \quad -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{i,j}}{t_i} \log \frac{c_{i,j}}{t_i} \tag{5.2}$$

By definition $0 \leq \epsilon_i \leq 1$, with equality if and only if $c_{i,j} = t_i$ and $c_{i,j} = t_i/N$, respectively. If ϵ_i is close to 1 it means that the word is present across the documents throughout the corpus, and if ϵ_i is close to 0 it means that it occurs only in a few specific documents. The global weight $1 - \epsilon_i$ is therefore a measure of the indexing power of w_i.

5.1.2 Singular value decomposition

The matrix $\mathbf{W}$ of size $(M \times N)$ resulting from the above feature extraction defines two representations for the words and documents. Each word w_i can be associated with a row vector of dimension N, and each document d_j can be associated with a column vector of dimension M. These vectors are sparse, and of large dimensions. The spaces spanned by them are distinctly different. Singular value decomposition (of $\mathbf{W}$) is performed retaining only the R largest singular values and their associated singular vectors. SVD maps the row and column vectors onto a smaller continuous space Ψ as follows:

$$\mathbf{W} \approx \hat{\mathbf{W}} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{5.3}$$

where, $\mathbf{U}$ is $(M \times R)$ left singular matrix with row vectors u_i $(1 \leq i \leq M)$,

$\mathbf{S}$ is $(R \times R)$ diagonal matrix of singular values $s_1 \geq s_2 \cdots \geq s_R > 0$,

$\mathbf{V}$ is $(N \times R)$ right singular matrix with row vectors $v_j (1 \leq j \leq N)$,

$R \ll \min(M, N)$ is order of the decomposition, and

T is matrix transposition.

We know that the matrices $\mathbf{U}$ and $\mathbf{V}$ are column orthonormal, which means that, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_R$, the identity matrix of R-dimension. Thus the column vectors of $\mathbf{U}$ and $\mathbf{V}$ each, define an orthonormal basis for the R dimensional space Ψ, spanned by the vectors u_i and v_j. The matrix $\hat{\mathbf{W}}$ is the best rank-R approximation to the matrix $\mathbf{W}$. The row vectors of $\mathbf{W}$ are projected onto the orthonormal basis formed by the column vectors of the right singular matrix $\mathbf{V}$. This defines a new representation of the words in this space. That is, the row vector $u_i\mathbf{S}$ (where multiplying by the diagonal matrix $\mathbf{S}$ is just a fixed scaling of each element) characterises the position of the word w_i in the underlying Ψ dimensional space, for $1 \leq i \leq M$. Similarly the column vector of $\mathbf{W}$ are projected onto the orthonormal basis formed by the column vector of the left singular matrix $\mathbf{U}$. This defines a new representation of the documents in Ψ. The row vector $v_j\mathbf{S}$ characterises the position of document d_j in R, for $1 \leq j \leq N$. We can call each of the M scaled vectors $\bar{u}_i = u_i\mathbf{S}$ as a *word vector*, uniquely associated with word w_i in the vocabulary, and each of the N scaled vectors $\bar{v}_j = v_j\mathbf{S}$ as the *document vector*, uniquely associated with the document d_j in the corpus.

The SVD mechanism defines a transformation between the high dimensional discrete entities (ν and $\mathcal{T}$) and a low dimensional continuous vector space Ψ, the R

dimensional space spanned by u_is and v_js. The matrix $\hat{\mathbf{W}}$ captures the major structural associations in $\mathbf{W}$, and ignores the higher order effects. The closeness between vectors is determined by the overall pattern in the language used in $\mathcal{T}$. If two words are close, they appear in similar documents conveying similar meanings.

5.1.3 Advantages and limitations of LSA framework

The latent semantic framework has interesting properties like

- Single vector representation for both words and documents in the same continuous vector space.
- An underlying topological structure reflecting semantic similarity.
- A well-motivated natural metric to measure the distance between words and between documents in the space.
- A low dimensionality that makes clustering meaningful and practical.

The feature of the LSA is that it is a bag of words approach. It does not take into account the order of the words in a sentence. It is suitable for capturing the semantics of the paragraph. But this does not take advantage of the syntactic and pragmatic constraints available. An alternative is to use word n-tuples that are formed by n successive words in the original document, and characterising the $\mathbf{W}$ matrix by co-occurrences of these n-tuples instead of words. Though the number of terms in the $\mathbf{W}$ matrix increases, it would still be feasible to use the LSA paradigm.

5.2 N-GRAM + LSA LANGUAGE MODELLING

A major application of the LSA framework is in statistical language modelling. Here it best works when it is applied in conjunction with the n-gram approach. To use

LSA for language modelling, we need a representation for the word and its history, a measure of closeness, and a method for computing probabilities.

The history (h_{q-1}) of a word w_q in a test document is defined as all the words occurring before the current word. The history is represented by the centroid ($\mathbf{m}$) of all the words it contains up to that point and is computed as:

$$\mathbf{m} = \sum_{i=1}^{q-1} \bar{u}_i \tag{5.4}$$

The cosine of the angle between $\bar{u}_q$ the representation of w_q and the centroid of the history is used as a measure of closeness. The cosine is computed between the LSA vector for the q^{th} word w_q in the document and the centroid ($\mathbf{m}$) of all the words in the history h_{q-1}.

$$\cos(\bar{u}_q, \mathbf{m}) = \frac{(\bar{u}_q \cdot \mathbf{m})}{\| \bar{u}_q \| \| \mathbf{m} \|} \tag{5.5}$$

To convert these cosine values into probabilities, the cosines are normalised between 0 and 1 by the following procedure. The smallest cosine c_h between the history and the word w_j which ranges over the vocabulary of M items is found.

$$c_h = \min_{j \in M} \cos(\bar{u}_j, \mathbf{m}) \tag{5.6}$$

The cosines are converted into probabilities by deducting c_h from all the cosines for a given history, and then normalising. As a first estimate, the LSA probability $\hat{P}^l(.)$ of the word w_q for the history h_{q-1} is obtained as:

$$\hat{P}^l(w_q|h_{q-1}) = \frac{\cos(\bar{u}_q, \mathbf{m}) - c_h}{\sum_{j=1}^{M} [\cos(\bar{u}_j, \mathbf{m}) - c_h]} \tag{5.7}$$

The superscript l refers to LSA model.

If the vocabulary is large, then the probability values will hover around $1/M$, not differing enough to contribute significantly. The dynamic range of the LSA probability $\hat{P}^l(.)$ is increased by raising it to some power (γ) and renormalising it.

$$P^l(w_q|h_{q-1}) = \frac{\hat{P}^l(w_q|h_{q-1})^\gamma}{\sum_{j=1}^{M} \hat{P}^l(w_j|h_{q-1})^\gamma}, \quad (5.8)$$

where, the value of γ needs to be determined empirically.

5.3 INTEGRATION WITH N-GRAM MODEL

The LSA model is a good predictor of content words, while the n-gram model is good at predicting the immediate context. It is desirable that at least half the probability value predicted is contributed by the n-gram model. This is achieved by discounting the LSA probabilities. A confidence measure λ_q, derived from (5.2) for normalised entropy ϵ_q, is used to discount the LSA probabilities [7]. The confidence measure is defined as $\lambda_q = (1 - \epsilon_q)/2$. A high value of $1 - \epsilon_q$ implies that the word w_q occurs less frequently and is likely to be a content word. In this case, the LSA model will predict the probability better. In such cases, when λ_q is high, it ensures that the probabilities from both the LSA and n-gram model is considered equally (5.9). A low value of $1 - \epsilon_q$ implies that the word w_q occurs frequently in the data. In this case, the n-gram model predicts the word better. Here, λ_q gives a higher weightage to the n-gram probability.

The LSA model often predicts words that are syntactically disallowed. A linear combination with the n-gram model does not help. It is desirable to use a nonlinear combination function that gives higher probability when the two models agree, i.e., the word prediction is both syntactically and semantically plausible, and gives a low probability if either model considers the word implausible. The geometric mean is the nonlinear combination function that seems to work well. The probabilities from the LSA model (P^l) and the n-gram model (P^n) can be combined as follows:

$$P(w_q|w_1, w_2, \cdots, w_{q-1}) = \frac{P^l(w_q|w_1, w_2, \cdots, w_{q-1})^{\lambda_q} P^n(w_q|w_{q-1})^{1-\lambda_q}}{\sum_{j=1}^{M} P^l(w_j|w_1, w_2, \cdots, w_{j-1})^{\lambda_j} P^n(w_j|w_{j-1})^{1-\lambda_j}} \quad (5.9)$$

The performance of the language models is measured using perplexity. If Q is the total number of words in the test set, the perplexity (β) of the model in (5.9) is given by:

$$\beta = \exp\left(-\frac{1}{Q}\sum_{q=1}^{Q}\log P(w_q|h_{q-1})\right) \tag{5.10}$$

where h_{q-1} is the history $(w_1, w_2, \cdots, w_{q-1})$ of the word w_q.

5.4 DATABASE OF NEWS STORIES IN TAMIL

In LSC type of lanaguage modeling it is desirable to have text documents pertaining to one semantic content like a news story or a passage. A large text corpus of news bulletins is not available for most Indian languages. To realise this, the transcription pertaining to the speech corpus as manually categorised into 8 different story categories, namely, economics, events, others, politics, sport, war, weather and world politics. The categorisation also indicates the diversity of the database. The description of the database in terms of news stories and the number of words in the news stories is given in Table 5.1 and 5.2, respectively. The corpus is partitioned into a training set and test set. Four news bulletins were used for testing and the rest for training. The categories of stories that appear in these bulletins is shown in Table 5.1.

5.5 WORD-BASED BIGRAM LANGUAGE MODEL

Due to the small size of the text corpus, deriving trigram language models is not appropriate. A limited vocabulary of 1277 words which have at least 4 occurrences in the database is chosen. A bigram language model at the word level is derived using the CMU toolkit [148] for that corpus. The performance of the bigram language model is determined in terms of perplexity using (5.10). For the computation of the perplexity of the bigram model, h_{q-1} reduces to w_{q-1}. The perplexity of the bigram model for

Table 5.1 Description of the Tamil news database in terms of stories.

Story category	# documents	
	Training set	Test set
Economics	44	2
Events	104	23
Others	124	24
Politics	104	4
Sports	34	3
War	163	32
Weather	6	1
World politics	64	3
Total	643	92

the test set is 233.

5.6 WORD-BASED BIGRAM + LSA LANGUAGE MODEL

The latent semantic analysis language model is derived from the training corpus using the procedure detailed in Section 5.2. This model is combined with the bigram language model to derive the large span constraint (bi-LSA) language model. One important step in developing the language model is the dimensionality reduction by means of SVD. The optimal dimension to be used has to be determined. The dimensionality reduction is done to project the words onto a lower dimensional space Ψ . Based on some suitable distance metric, the semantic similarity between two words in Ψ is greater, if the distance between them is small.

Table 5.2 Statistics of the database of news stories.

	Training set	Test set
Total # documents	643	92
Total # words	26380	3706
Min. # words/story	6	8
Max. # words/story	159	136
Avg. # words/story	41	40

The optimum dimension is not known *a priori* [98]. In the studies conducted, the dimensionality reduction by SVD is varied to obtain the best results.

A matrix $\mathbf{W}$ is created as described in Section 5.1.1. There are 1278 words in the vocabulary. The training set comprises of 643 documents. The cell $w_{i,j}$ contains the weighted count of the number of times word i occurred in the document j. The SVD is performed on this $\mathbf{W}$ matrix. The matrix $\mathbf{S}$ of singular values is truncated to different orders of decomposition. The corresponding singular vectors of $\mathbf{U}$ and $\mathbf{V}$ are retained. The LSA model based on (5.8) is derived. This model is combined with the bigram language model. The perplexities of both the bigram language model and the combined bi-LSA language model for truncation of SVD to order 200 is given in Table 5.3. The optimal performance is obtained for a truncation order of 200. From the table we observe that the combined bi-LSA model performs better than the bigram model as expected. The perplexity is reduced by 64.3% when the combined large span constraint model is used, as compared to the bigram model. Another factor in (5.9) is the value of λ_q. It depends on the frequency of occurrence of the word w_q in the database and varies for every word as shown in Equation (5.9). If a word is infrequent

Table 5.3 Perplexity values for the word-based bi-LSA model for the test set where the SVD is truncated to order 200.

Language model	Perplexity
Bigram	233
LSA	536
bi-LSA	84.3

it may be a content word. For these words λ_q will be high, closer to 0.5. In such cases both the bigram and the LSA model will have about equal weightage in the combined model. Conversely for frequent words the entropy will be high and λ_q low. Then more weightage would be given to the bigram model. The effect of λ_q on all the different SVD truncation orders and membership thresholds is uniform.

5.7 SYLLABLE-BASED LARGE SPAN CONSTRAINT LANGUAGE MODEL

The syntactic and semantic constraints associated with a word were exploited in the previous section to develop a large span constraint model. Syllables are sound units smaller than words. In spoken language, arbitrary sequence of syllables is not permissible, as no meaningful message can be then conveyed. Hence there exist some constraints at the level of syllables. The contextual constraints in the syllable sequence can be determined using a standard bigram language model using syllables as the basic unit. If the training corpus is parsed into syllables, then a standard language modelling toolkit [148] can be used to derive the bigram language model at the level of syllables. One such model was used in the syllable recognizer in Section 4.3 to improve its performance.

Syllables are not normally associated with semantics. If two syllables co-occur in

a paragraph describing some topic of discourse, it cannot be directly inferred that the syllables are related to each other. But there exists some constraints of the language that precludes some syllables following a syllable in certain context. It is desirable to model these latent constraints to achieve an improved performance in language modelling.

In large passages of text like a paragraph, the number of syllables is large. In such cases, the co-occurrence constraints if any, are likely to be masked. If there exist some weak constraints between co-occurring syllables, it is more likely to be manifested when the document size is small.

The latent constraints among the syllables is modelled in the LSA framework. The database in table 5.1 is parsed into syllables. A vocabulary of syllables is chosen similar to the vocabulary of words in the large span language models. Using an arbitrary threshold of at least 4 occurrences of a syllable in the Tamil database, the vocabulary of 1044 syllables is derived. For this vocabulary, the optimum number of syllables per document and the truncation order of the SVD decomposition that yields best performance need to be determined. Strings of nonoverlapping syllables of varying length are considered as the training documents.

To better exploit the latent dependencies among the syllables, if any, the syllable-based LSA and bigram models are combined. This syllable-based bi-LSA language model is used for evaluating the test set. The perplexity of the bi-LSA language model is determined for training documents of different lengths and different orders of SVD truncation, as given in Table 5.4. The perplexity increases marginally as the number of syllables in the document increases. This is because, for larger document sizes the syllables are co-occurring with many other syllables, and the model is unable to capture the latent constraints, leading to poorer performance. The table also shows

Table 5.4 Perplexity values for the syllable-based bi-LSA model for different SVD orders and various sizes of the documents in training (Perplexity for the syllable level bigram language model is 29.3).

SVD order	# syllables per document				
	5	10	20	75	300
75	23.7	23.7	24.2	24.9	25.5
100	**23.1**	23.7	24.2	24.8	25.5
150	23.9	23.9	24.2	24.8	25.5
200	24.1	23.9	24.2	24.8	25.5
250	24.3	23.9	24.2	24.9	24.8

that for syllable-based large span language models, the truncation order of SVD does not seem to influence the performance greatly as the size of the database is small. The best performance (perplexity of 23.1) is observed for documents of size 5 syllables, where the order of truncation of the SVD is 100. The corresponding perplexity of the bigram model for the test set is 29.3.

We have shown that there may be some latent constraints in the co-occurrences of some syllables, and that they can be modelled using the large span language models. The reduction in perplexity of this model over the standard syllable level bigram model is 22%. If such a model is incorporated into a syllable recognizer, the performance of that recognizer is likely to improve further.

5.8 SUMMARY

In this chapter, we reviewed the basic concepts of LSA and its adaptation for use as a language model. A database of news stories, suitable for LSA studies, was developed

from the speech corpus. An LSA-based large span constraint language model was developed at the word level for the Tamil language. This model was combined with a standard bigram language model to obtain an improvement in the performance of the language model. We showed that latent large span constraints exist at the syllable level also. A syllable level large span constraint model was combined with the standard syllable-based bigram model. The performance of this syllable-based bi-LSA model was studied for various syllable level document sizes and orders of SVD truncation. The performance of this model was poorer when the number of syllables in a document was increased. Conventionally bi-LSA language models are developed using large databases comprising of millions of documents and vocabularies of about 10,000 words. The order of The order of SVD truncation has a significant effect on the performance of those models. In the speech-based bi-LSA model, the size of the database used is small. The order of SVD truncation does not seem to affect the performance of the model. Overall, the performance of the bi-LSA model was better than the syllable level bigram model by 22%. The summary of the issues discussed in this chapter is given in Table 5.5. In the next chapter, we discuss the effect of erroneous transcripts (simulating errors due to speech recognition) on the large span constraint models.

Table 5.5 Summary of techniques to use LSA for language modelling.

- There exist semantic constraints among the words in a paragraph or news story. These constraints are spread over large spans. If these constraints are incorporated into a speech system the performance is likely to improve.
- The extension of LSA for language modelling was recollected. A text-based bi-LSA model was created for Indian languages.
- It was shown that large span constraints seem to exist at the syllable level also. The syllable level text-based bi-LSA model was developed and shown to be better than the syllable level bigram model.

CHAPTER 6

LARGE SPAN CONSTRAINT LANGUAGE MODEL USING ERRONEOUS TRANSCRIPTS

In certain situations, the erroneous transcriptions from speech recognition systems may be the only text available for developing language models. For example, in dialogue modelling applications, no database of textual dialogues is available. The text corpus would be derived from speech transcripts. For many languages, especially non-literary languages, large text corpus may not be available. For these languages, it is easy to record speech data in large volumes. In such situations, it may be required to derive a language model directly from the speech signal. In all such cases, the effect of the errors, introduced due to decoding, on the generation of a language model needs to be studied. In this chapter we study the effect of noisy text input on large-span language models.

6.1 TYPES OF ERRORS IN SPEECH TRANSCRIPTION

In speech recognition, errors are normally of three different types, insertion, deletion and substitution. Various combinations of these errors are simulated by altering the matrix of row counts $\mathbf{W}$ before normalisation. The effects of different combinations of these errors are studied. The errors of the speech recognition system result in the production of erroneous transcriptions. When these erroneous transcripts are used to develop a large span constraint language model, they may affect its performance.

Insertion errors occur when an extra word is inserted in the sentence recognition hypothesis, leading to more number of words than those actually present in the spoken

utterance. If the output of the speech recognizer is aligned with the reference transcript for maximal matching, we may find that some words are deleted or are not recognized in the output. These errors are termed as deletion errors. Due to the confusability among the words in the vocabulary and the variability in speech, a particular word is misrecognized as some other word close to it. These types of errors are termed as substitution errors. The substitution errors are generally more frequent than the other two types of errors. These three types of errors do not occur in isolation. While developing the recognizer, generally the parameters of the recognizer are so adjusted that the insertion and deletion errors are minimal. In simulating the effects of the errors, we have limited our scope to simulating erroneous source transcripts to be used by the large span constraint model. This is done, as we are interested in determining the effect of the errors in recognition on the language model, when it is derived directly from the speech signal as detailed in the next chapter. In this study, the errors are simulated in the transcripts used to build the large span constraint model. The bigram model is trained on a clean text (as used in the previous chapter). These two models are combined to obtain the bi-LSA model.

6.2 INSERTION ERRORS

Erroneous transcripts with insertion errors are simulated as follows. Let $\tilde{\mathbf{W}}$ be the matrix of raw counts. Each column of the matrix contains the frequency counts of the words in the vocabulary for a document in the database. That is, the number of columns of $\tilde{\mathbf{W}}$ correspond to the number of documents in the database, and the rows correspond to the number of words in the vocabulary. These are raw counts before being normalised for the entropy and the length of the document. A sparse random matrix $\mathbf{N}_I$ of ones, of the same size as $\tilde{\mathbf{W}}$, and of desired sparseness (say 95%) is created. The density (defined as [# of nonzero elements of the matrix]/[product of the

dimensions of the matrix]) of this matrix $\mathbf{N}_I$ (1-0.95) is the amount of insertion errors (e_I) we desire to simulate, i.e. 5%. The matrix $\mathbf{N}_I$ is added to $\tilde{\mathbf{W}}$ to give us $\tilde{\mathbf{W}}_I$. $\tilde{\mathbf{W}}_I$ now contains 5% insertion errors at random locations. This matrix is weighted by the entropy of the word and normalised for the length of the documents. The error simulated $\tilde{\mathbf{W}}_I$ is used in the place of $\mathbf{W}$ in Section 5.2 for developing the integrated bi-LSA model. The simulation of errors is carried atleast 5 times and the average values are reported in the tables of this chapter. The performance of this model is determined for the test set. The performance is evaluated for different insertion error rates, and is given in Table 6.1. The density of the matrix $\tilde{\mathbf{W}}_I$ for different error rates is also given. As the insertion rate increases, the density of the matrix $\tilde{\mathbf{W}}_\mathbf{I}$ increases rapidly and the perplexity reduces marginally. Because the values of the components of the vectors in the reduced dimensional space Ψ increases, the cosine measure results in a higher prediction probability for words close by and thus a marginal reduction in perplexity.

Table 6.1 Perplexity values for the combined bi-LSA model derived from erroneous transcripts for various insertion error rates e_I. The truncated SVD is of order 200.

e_I (%)	Density of $\tilde{\mathbf{W}}_I$	Perplexity
05	6.56	83.38
10	11.1	81.43
15	15.5	81.00
20	19.6	80.45
35	30.8	78.94
50	40.5	78.26

6.3 DELETION ERRORS

When a deletion occurs in a word string, the resulting sequence of words may be linguistically incorrect. If a n-gram language model is trained on such erroneous transcripts, its performance would be poor. As the large span constraint model ignores the word order, sequencing information is not important. However, this model is a good predictor of content words. If the function (frequently used) words are deleted, the performance of the large span constraint model is unlikely to be affected. In contrast, if the content words are deleted, the performance of the model may suffer.

To simulate the deletion errors, a sparse random matrix $\mathbf{N}_D$ of ones is created as in the previous section. This matrix is subtracted from $\tilde{\mathbf{W}}$ only where positive counts exist to give $\tilde{\mathbf{W}}_D$. The elements of $\tilde{\mathbf{W}}_D$ are then weighted by entropy and normalised for the lengths of the documents. As mentioned in the previous section, the matrix $\tilde{\mathbf{W}}_D$ is used in the place of $\hat{\mathbf{W}}$ to derive the probabilities of the LSA model. The performance of the language model for different deletion rates is shown in Table 6.2. As the original matrix $\mathbf{W}$ is sparse, the deletion errors cause only a small reduction in the density of $\tilde{\mathbf{W}}_D$. We observe from Table 6.2 that as the deletion error rate increases, the performance of the combined bi-LSA language model reduces, with the worst performance when the error rate is 35%. One reason for the fall in performance could be that, when the rate of the deletion errors is high, it is more likely that content words are deleted.

6.4 SUBSTITUTION ERRORS

Substitution errors generally occur between words with similar acoustic realisations. They can occur due to poor training of the models, and large out of vocabulary rates. This error is simulated by a deletion followed by an insertion as follows: A sparse

Table 6.2 Perplexity values for the combined bi-LSA model derived from erroneous transcripts for various deletion error rates e_D. The truncated SVD is of order 200.

e_D (%)	Density of $\tilde{\mathbf{W}}_D$	Perplexity
05	1.70	78.16
10	1.63	73.42
15	1.56	76.43
20	1.45	72.28
35	1.32	74.38
50	1.17	73.82

random matrix N_s of the desired sparseness is created. If there is a non-zero count in $\tilde{\mathbf{W}}$ at locations corresponding to the location of ones in N_s, the counts are reduced by one. Depending on the number of deletions in a column vector of $\tilde{\mathbf{W}}$, the same number of ones are added back to the column vector of $\tilde{\mathbf{W}}$ at some random locations to obtain the matrix $\tilde{\mathbf{W}}_S$. The density of the matrix $\tilde{\mathbf{W}}$ would vary, depending on the locations of addition and deletion of counts. If the deletion occurs in locations having single counts but additions occur in locations with previously non-zero counts, then the density will reduce. Similarly other situations exist which would modify the density of the matrix $\tilde{\mathbf{W}}_\mathbf{S}$.

The performance of the bi-LSA model for different substitution errors is shown in Table 6.3. The perplexities of this model at 10% substitution error is the least. But as the error rate increases, the perplexity also increases. In all these trials on error simulations it must be noted that a 20% error does not mean that actually 20% of the words were deleted/substituted. This 20% error would occur when the matrix $\tilde{\mathbf{W}}$ is fully dense. Thus the actual errors introduced into the matrix would be less than

that simulated due to the sparse nature of the matrix. This can be inferred from the density of the matrices. For visualisation the three different types of errors are plotted in Figure 6.1.

Table 6.3 Performance of the combined bi-LSA model derived from erroneous transcripts for various substitution error rates e_S. The truncated SVD is of order 200.

e_S (%)	Density of $\tilde{\mathbf{W}}_S$	Perplexity
05	1.71	86.28
10	1.64	82.30
15	1.60	85.51
20	1.52	86.30
35	1.36	88.68
50	1.32	90.40

6.5 COMBINATION OF RECOGNITION ERRORS

In the transcription from a speech recognizer, the insertion, deletion and substitution errors occur in some combination. Different combinations of speech recognition errors are simulated. To study the effect of all three errors together, two type of errors were kept constant while varying the third. The Table 6.4 shows the effect of different combinations of all the errors encountered. As expected the perplexities for the test set is higher for the combination of insertion, deletion and substitution errors as compared to their individual occurrence. But the overall perplexity of the bi-LSA model trained with erroneous transcripts is about the same as that trained with error-free transcripts. Thus, it is unlikely that speech recognition errors will severely degrade the performance of a large span constraint model derived from the speech signal.

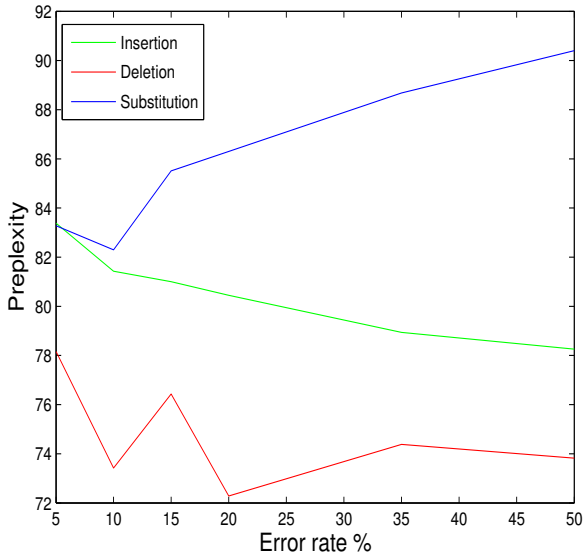

Fig. 6.1 Perplexity of the test set for various levels of insertion, deletion and substitution errors.

Table 6.4 Perplexity values for the combined bi-LSA model derived from erroneous transcripts for various combination of insertion, deletion and substitution error rates. The truncated SVD is of order 200.

e_S (%)	Perplexity		
	$e_I = 5\%$ $e_D = 5\%$	$e_I = 10\%$ $e_D = 5\%$	$e_I = 10\%$ $e_D = 10\%$
05	83.9	81.5	82.0
10	84.4	82.2	82.4
15	83.9	82.4	83.0
20	83.6	82.8	81.6

6.6 SUMMARY

Towards developing speech-based large span constraint models, one of the main issues that needs to be addressed is the likely errors at the recognition stage. To understand the effect of these errors on the large span constraint models, the commonly occurring insertion, deletion and substitution errors were simulated, in isolation and in combination. It was observed that at a maximum of 20% error rate, the performance of the bi-LSA model trained on erroneous transcripts gives about the same performance as that of the model trained on error-free transcripts. This suggests that a speech-based bi-LSA model is unlikely to be greatly affected by moderate levels of recognition errors. The issues discussed in this chapter is summarised in Table 6.5. In the next chapter, we discuss the development the speech-based bi-LSA model.

Table 6.5 Summary of studies on developing large span constraint models from erroneous transcripts.

- In many situations, error-free text corpus in electronic form may not be available. Either erroneous transcripts from a speech recognizer, or speech data may only be available. Previous studies do not reveal the effect of erroneous (transcripts) input to the language model.
- In this chapter, the effect of erroneous transcripts on the large span constraint model, for different errors, was studied.
- Insertion, deletion and substitution errors of up to 20%, in isolation or in combination, do not seem to affect the performance of the large span constraint model.
- It was thus inferred that, the performance of the large span constraint model developed directly from the speech signal may not be severely affected.

CHAPTER 7

SPEECH-BASED LARGE SPAN CONSTRAINT LANGUAGE MODEL

7.1 INTRODUCTION

In the previous chapters, the use of latent semantic analysis (LSA) to capture the global semantic constraints, and bigram models to capture local syntactic constraints, were shown to reduce the perplexity of the test set for the text-based bi-LSA model. In this chapter, we propose approaches to derive LSA-based large span constraint language models directly from the speech signal. A reference segment is derived from the speech signal for each word in the vocabulary. Based on the normalised distance between the reference word segment and the word segment in the training data, the LSA model is derived. We show that this speech-based bi-LSA model in combination with the standard bigram model performs better than the conventional text-based bi-LSA model. The results are demonstrated for a limited vocabulary on the Tamil database.

One of the main applications of statistical language models is in speech recognition. The use of speech knowledge, prosodic constraints, large span semantic and local syntactic constraints, when integrated with the speech recognizer would improve its performance. We propose a method in which the semantic constraints, in terms of the co-occurrence of words in a document, can be captured indirectly from the speech signal in the framework of latent semantic analysis. We show that the speech-based large span constraint model performs better than the bigram model, and the text-based bi-LSA model. The reduction in perplexity for a test set is used as a measure of

performance of the model. In the following sections, we give details of the construction of the matrix **W** from the acoustic signal. Using this speech-based matrix **W**, we develop the speech-based bi-LSA model.

7.2 DEVELOPMENT OF THE MATRIX W

The first step in developing the large span constraint model is to construct the matrix **W** from the speech signal. Given the speech signal, the objective is to determine the word boundaries in the speech signal and recognize the word present in the speech segment. The ideal situation would be to use a speech recognizer to recognize the sequence of words present in the speech signal. The need for the language model is to improve the performance of the speech recognizer as it is poor. In most cases a good speech recognizer may not be available. We explore two basic approaches in pattern recognition for the task, namely, the Dynamic Time Warping (DTW) approach and the template matching approach to identify a word in the segment of speech, and also to find the other closest matches. This information is used in deriving the matrix **W**.

7.2.1 Dynamic time warping approach

Dynamic programming is used in speech processing applications for time alignment and normalisation to compensate for variability in speaking rate in template based systems [11]. Let us consider two speech patterns A and B representing the spectral vector sequences $(\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_i}, \cdots, \mathbf{a_M})$ and $(\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_j}, \cdots, \mathbf{b_N})$, respectively, where $\mathbf{a_i}$ and $\mathbf{b_j}$ are parameter vectors of short-time acoustic features. Dissimilarity or distance between the two sequences A and B for a particular path ϕ is given by $d_\phi(A, B)$. Time normalisation of A and B is obtained by finding the best temporal match given by the minimum dissimilarity $d(A, B)$, defined as:

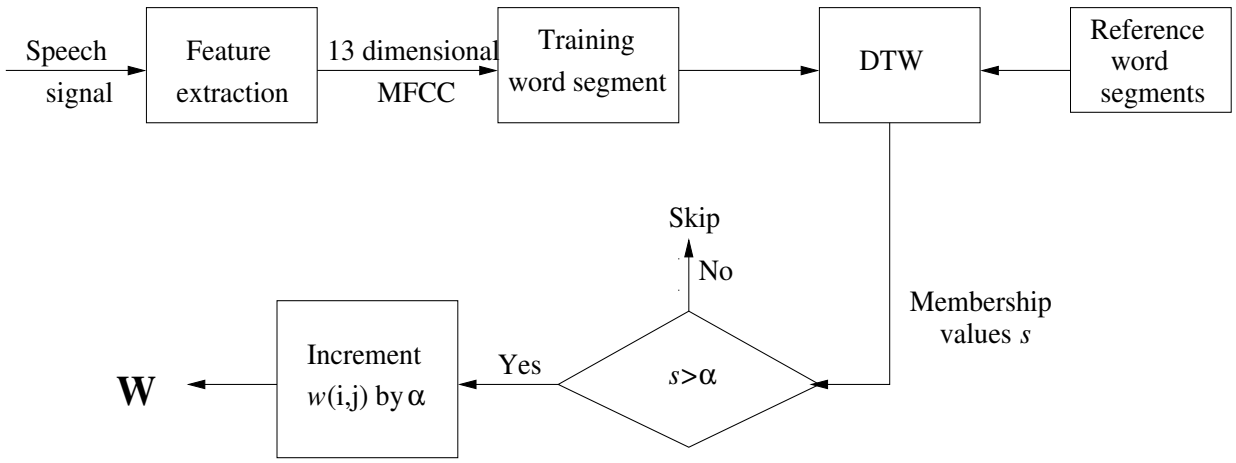

Fig. 7.1 Block diagram for construction of **W** in the proposed speech-based LSA language model using DTW.

$$d(A,B) \;=\; \min_{\phi}(d_{\phi}(A,B)). \tag{7.1}$$

When A and B represent segments of two utterances of the same word class, the choice of the best path implies that the dissimilarity is measured using the best possible alignment between the different regions of the two segments. In dynamic time alignment, a set of local continuity constraints is imposed on the warping function, which does not result in the omission of any important information bearing events in the speech signal. The definition of the dissimilarity measure given by (7.1) involves a minimisation process that can be effectively implemented by dynamic programming. The optimal warping path between two word segments provides the best match between the segments. The dissimilarity (distance) gives the measure of how close the two word segments are to each other.

The objective is to fill up the matrix **W** with counts if a word in the vocabulary is present in a document. In the speech context, the document is a set of spoken utterances (speech file). We need to find if a word in the vocabulary is present in the

spoken document or not. If present, the corresponding element of **W** is incremented. To identify the words in a speech utterance using dynamic time warping, word segments are required. The knowledge of the word boundaries is known, as the Tamil database is segmented in terms of words and syllables. Fig. 7.1 shows the block diagram of the DTW approach to construct the matrix **W** from the speech signal. To perform DTW, a reference segment is required for each word in the vocabulary. The acoustic realisations of the words vary largely. Hence an appropriate reference segment for each word in the vocabulary is chosen manually. From the speech signal 13 dimensional MFCC vectors are derived for each frame of 20 msec using a 10 msec shift, for every word segment in the speech document, and for the reference word segments. Given a speech document and the word boundaries, in the model building stage for every word segment in the document, the DTW alignment between that word segment and all the valid reference word segments in performed. The path constraints in DTW restrict the length of the test segment to be in the range of half to twice the length of the reference segment. Thus a word segment in the speech document is matched only with those reference segments that fall within this range. The result is a distance value between the word segment in the speech document and each reference pattern in the vocabulary. The distances are then normalised between 0 and 1, and are denoted by d_n. Ideally the reference word with the least distance would be the choice for the closest match, and the count in the corresponding cell of matrix **W** would be incremented. Instead we define a membership (s_i) based on the normalised distances (d_{n_i}). The normalised distance d_{n_i} denotes the dissimilarity between a word pattern vector and the i^{th} reference pattern. Thus $s_i = 1 - d_{n_i}$ is the membership of a word pattern to the i_{th} reference pattern. It indicates how close a reference word segment is to the training/test word segment. If the membership is above a certain threshold

(α) then the appropriate element $w_{i,j}$ is incremented by the membership value. The elements of **W** are also scaled for the length of the document (# words) and weighted by the entropy of the term. This procedure is carried out for each word segment in the document, and for all the documents in the database, thus deriving the matrix **W** from the speech signal.

7.2.2 DTW-based bi-LSA model

For the Tamil database a vocabulary of 1277 words is chosen (words with at least 4 examples in the database). The matrix **W** is constructed using the procedure detailed in the previous section. The DTW is performed between every word segment of a document in the training set and all the reference word segments of similar durations. The membership threshold, for deciding on the closeness of a word, is varied. Similarly, the truncation order of the SVD is also varied. For different values of these parameters, the combined bi-LSA model is derived from the speech signal. For the test set the perplexities obtained are tabulated in Table 7.1. From the resulting perplexities it is observed that the model derived from the speech signal using DTW performs better than the bigram model (perplexity 233) and the text based bi-LSA model (perplexity 84.3). The reduction in perplexity over the text based bi-LSA model is 7%.

7.2.3 Template matching approach

The approach described in the previous section used DTW. The inherent drawbacks of DTW is that it is speaker dependent. It does not work well when large variations in the speaker characteristics are observed. However, this is not a limiting factor in the DTW-based approach, since we are not interested in the best match, but a few closely matching words. In the DTW approach, there is a limitation on the length of the reference/test segment. That is, segments of arbitrary lengths cannot be matched.

Table 7.1 Perplexity values for the speech-based combined bi-LSA model derived using DTW approach for various membership threshold values (Perplexity for the text-based bi-LSA model was 84.3).

SVD truncation order	Membership threshold α			
	0.85	0.96	0.97	0.98
75	79.13	80.8	80.83	80.94
100	79.30	81.2	81.73	81.00
150	79.11	81.4	82.20	80.47
200	**79.02**	82.0	82.59	82.42
250	79.32	83.1	83.85	82.57

Another problem is that every word segment in the training corpus needs to be matched with all the reference patterns in the vocabulary. When the vocabulary and the training corpus is large, the time consumed in developing the model is prohibitively expensive. Hence an alternate approach is desirable. In this section we discuss a template-based approach. In the template-based approach, a fixed dimensional representation for each word segment is defined. A reference template for each word in the vocabulary is constructed using this word representation. A similar fixed dimensional representation, called word pattern vectors, is determined for each word in the training documents. The similarity between the reference template and each word pattern vector is used to derive the matrix $\mathbf{W}$ using the same procedure as described in the DTW-based approach. The main advantage of this approach is that, the template for a word needs to be constructed only once. This reduces the time consumed for matching the reference template with the word pattern vectors in the model building or testing stage as it involves computation of the distance measure only once.

The block diagram of the proposed template-based approach to develop the matrix **W** from the speech signal is shown in Fig. 7.2. The first step is to derive fixed dimensional pattern vectors to represent a word. This is achieved as follows: From the speech signal of a word segment, 13 dimensional MFCC vectors are derived for each frame of 15msec using a 1msec shift. A small frame size and frame shift is used to minimize the dissimilarity between the adjacent frames. The Euclidean distance between the adjacent pairs of feature vectors is determined for the entire sequence of feature vectors in the word segment. Based on the average duration of the word in the database, the number of frames required to represent a word is determined. The feature vectors corresponding to the desired frames are concatenated to form a pattern vector that represents a word segment. If the number of frames in the word segment is less than that required to construct the word pattern, then the frame with the smallest Euclidean distance with its neighbour is replicated. For a word segment where the number of frames extracted is larger than desired, the frame with the least Euclidean distance to its neighbor is dropped. This procedure is repeated until the desired number of frames for a word segment is obtained. It is assumed that there is minimal distortion/loss in adding/dropping the above frames. The selected frames are concatenated to form a fixed dimensional pattern vector representing the word. The resulting pattern vector has large dimensions. Comparing pattern vectors in such a high dimension space is not preferable. It has been shown that non-linear compression of large dimensional pattern vectors of speech using Auto Associative Neural Networks (AANN) models does not degrade the speech recognition performance [155]. In this study AANN models are used to compress the large dimensional pattern vector to around 60 dimensions. Thus a reduced dimension pattern vector is derived for each word segment in the entire training set. All the (reduced dimension) pattern vectors

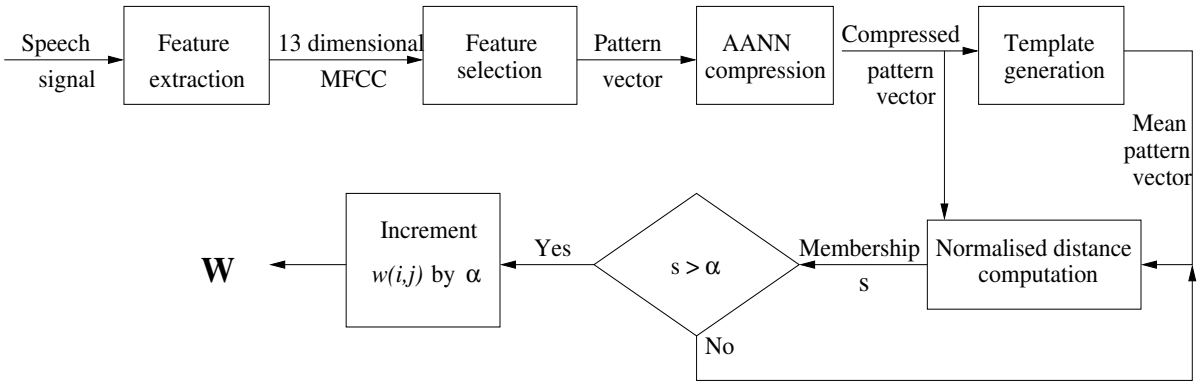

Fig. 7.2 Block diagram for construction of **W** in the proposed speech-based LSA language model using template matching.

corresponding to a word in the training set are used to derive a mean pattern vector, which serves as the reference pattern vector for that word in the vocabulary. One such reference pattern vector is derived for each word in the vocabulary.

For every word segment in a training document (speech file), a word pattern vector is derived as explained. The Euclidean distance between this pattern vector and all the reference pattern vectors in the vocabulary is determined. The resulting distances are normalised between zero and one. As in the previous section, the membership is defined as $s_i = 1 - d_{n_i}$, where d_{n_i} is the normalised distance. The membership indicates how close the word pattern vector is to each of the reference pattern vectors in the vocabulary. If this membership is above a certain threshold α, then the appropriate element $w_{i,j}$ is incremented by the membership value. The elements of **W** are also scaled for the length of the document and weighted by the entropy of the term. Thus the **W** matrix is derived from the acoustic signal using the template-based approach.

7.2.4 Template-based bi-LSA model

In this section, we first discuss the actual development of the matrix **W** using the template-based approach as outlined in the previous section. The average duration

of the words in the database is found. Based on this, a compact representation of 30 frames for each word is chosen. The 13 dimension MFCCs corresponding to each of the 30 frames are concatenated to obtain a 390 dimensional pattern vector. The 390 dimensional pattern vector is chosen, since it performed better than pattern vectors of various other dimensions experimented with. This pattern vector is nonlinearly compressed to 60 dimension using an AANN model. Other compact dimension are also explored. The structure of the AANN model used is $390L \quad 585N \quad 60N \quad 585N \quad 390L$, where L refers to a linear unit and N to a nonlinear unit, the numbers represent the number of nodes in a layer. The AANN model was trained for 200 epochs using all the pattern vectors in the training data. The compressed vectors are obtained from the compression layer of the AANN. This compressed 60 dimensional pattern vector is the word pattern vector. Such pattern vectors are derived for each of the word segments occurring in the training data. These word pattern vectors are used to construct the matrix **W** as described previously.

The performance of the speech-based bi-LSA model derived using the template-based approach is given in Table 7.2 for different membership threshold values, and various SVD orders.

Different SVD orders are used to determine the optimal dimension that captures the semantic relatedness better. If the threshold is high (say 0.98) the **W** matrix is similar to its text based counterpart in its sparseness. As the threshold is lowered, more elements of the **W** matrix are filled, which is like smoothing. The performance of the model improves. For low thresholds the performance is likely to deteriorate.

When DTW or template matching approach is used to recognize words, the correct word may not always be the best match. If the top few matching words are considered by defining a closeness measure, then the procedure will result in the acoustically clos-

Table 7.2 Perplexity values for the speech based combined bi-LSA model derived using template-based approach for various membership threshold values and SVD truncation orders. The word pattern vectors are compressed from 390 to 60 dimension using AANN models.

SVD truncation order	Membership threshold			
	0.92	0.96	0.97	0.98
75	78.0	78.3	77.8	79.2
100	77.8	78.4	78.2	78.9
150	**77.5**	78.9	78.3	79.9
200	78.0	79.5	79.4	80.7
250	78.6	80.1	78.2	81.3

est set of words being considered. The word counts corresponding to these words will be incremented. The bi-LSA model will learn the relationship among these acoustically confusable words. As an illustration the template based approach is executed using a membership threshold of 0.92. Shown below are three words and their closest matches as determined by the threshold. The membership values are shown in parenthesis. We see that in this approach at times there is an exact match, no match or many possible matches.

Reference word is not the first match.

inda−− > ellA (0.993) enda (0.987) inda (0.965) indap (0.968) indiyA (1) nalla (0.930) piññar (0.932)

Reference word is the exact match.

sadavihida−− > sadavihida 1

Reference word has no match among the top few.

peRRadu−− > piRahu (0.935) terivittañar (1) tiRandu (0.959)

In the speech-based bi-LSA model, using only the 1-best match is not desirable. The primary reason can be seen from the matches above. The 1-best in most cases may not be the correct word. Further when we increment counts for more number of words, it has a smoothing effect. In earlier studies word level and document level smoothing is shown to improve the performance of the bi-LSA model [8]. For the purpouse of illustration a sample document (column) of the speech-based and text-based matrices is shown in Table 7.3. It contains only the non-zero elements of **W**. As expected the speech-based **W** matrix has more non-zero entries than the text-based **W**. This will have a smoothing effect. If we measure the actual number of insertions, deletions and substitutions, as in Chapter 6 for the template-based approach we find theat the insertions are 262%, substitutions 25% and deletions near 0%. In the template-based bi-LSA model the 1-best match was used to construct the **W** matrix. The perplexity of the resulting bi-LSA model increased to 81.2 as against the previous 78 shown in Table 7.2. This shows that it is better to use soft counts rather than the 1-best match.

The performance in terms of perplexity of the test set for the three different language models is given in Table 7.4. The perplexity of the speech-based bi-LSA model is better than the standard bigram model by 66% and shows an improvement of 7% over the conventional text based bi-LSA model for an SVD order of 200.

7.3 SYLLABLE LEVEL SPEECH-BASED BI-LSA MODEL

In Chapter 5 we showed that a bi-LSA model can be derived at the syllable level using a text corpus. In this section, we develop a bi-LSA model from the speech signal at the syllable level.

Table 7.3 Non-zero entries in the speech and text-based bi-LSA raw matrices. The entries pertain to a single column (document 5) when a threshold of 0.92 is used.

S.No of word in vocab.	Member-ship	Weighted counts	S.No. of word in vocab.	Member-ship	Weighted counts	S.No. of word in vocab.	Member-ship	Weighted counts
2	0.92	0	378	0.93	0	932	1	0
12	0.93	0	403	1	0	934	0.92	0
23	0.97	0.01	418	0.94	0	953	0.94	0
25	0.96	0	451	0	0.00	956	1.97	0.01
29	1	0	466	0.95	0	958	1	0
35	0.95	0	470	1	0	960	0.93	0
45	0.99	0	471	1.91	0	962	0.92	0
46	1	0.03	496	0.94	0	967	0.95	0
47	0.93	0	499	0.93	0	972	0.96	0
48	1	0.01	501	0.94	0.01	1007	0.92	0
54	0.92	0	510	0.94	0	1014	0.92	0
88	0.97	0	512	0.98	0	1017	2.81	0
89	0.94	0	568	0.93	0	1031	0.96	0
100	0.92	0	574	0	0.01	1037	0.95	0
110	0.93	0	590	1	0	1041	0	0.01
112	0.92	0	660	0.97	0	1062	1	0
148	2	0.01	667	1	0.00	1063	0.93	0
150	0.92	0	672	0.97	0	1068	0.92	0
156	0.98	0	686	0.92	0	1076	1	0
157	0.98	0	687	0.92	0	1079	0.95	0
172	1	0	723	0.94	0	1087	0.97	0.00
173	0.92	0	726	0.96	0	1089	1	0
194	0.92	0.00	727	0.96	0	1092	0.99	0
197	0	0.01	728	1.92	0	1104	0.96	0
198	0.99	0	729	1.91	0	1107	0.95	0
199	1.91	0	743	0.97	0	1108	0.92	0
213	0.93	0	777	0.95	0	1112	0.96	0
222	1	0.01	780	1.87	0	1114	0.93	0
228	1	0	786	0.97	0	1117	0.93	0.01
233	0.93	0	787	0.98	0	1118	1.90	0
240	1.87	0.01	788	0.93	0	1136	0.94	0
242	0.92	0	790	0.96	0.01	1138	0.92	0
243	0.93	0	801	0.93	0	1157	0.98	0
244	1.93	0	808	0.94	0	1158	0.94	0
247	0	0.01	835	0.92	0	1167	1	0
265	0.92	0	836	1.86	0	1173	0.94	0
273	1	0.01	843	0.97	0	1191	0.96	0
277	0.99	0.00	844	0.93	0	1198	1.86	0
281	0.94	0	849	0.93	0	1201	0.94	0
282	0.98	0	860	0	0.01	1209	0.99	0
285	0.95	0	867	0.95	0	1213	0.95	0
291	1	0	869	1.87	0	1252	1.89	0
294	0.93	0	872	0.93	0	1268	0.96	0
296	0	0.00	876	0.93	0	1274	1	0
302	0	0.00	885	0.92	0	1276	1	0
308	0.93	0	891	0.95	0	1277	0.98	0
312	1.95	0	892	0.92	0	1278	0	0
318	0.99	0	900	1	0			
332	0	0.01	901	0.96	0			
337	0.93	0	924	0.95	0			
354	2.88	0	926	0.94	0			

Table 7.4 Comparison of performance of three different language models. All LSA models use a SVD truncation of order 200.

Model	Perplexity	Improvement over bigrams
Bigram	233	-
Bi- LSA	84.3	63.8%
Speech-based bi-LSA using DTW	79.0	66.0%
Speech-based bi-LSA using template matching	78.0	66.5%

Table 7.5 Perplexity values for the test set for three different language models at the syllable level. The speech-based bi-LSA model uses the DTW approach. All LSA models use an SVD truncation order of 200.

Language model	Perplexity
Bigram	29.3
Text-based bi-LSA	23.1
Speech-based bi-LSA	24.1

Segments of syllables are required for the development of the speech-based bi-LSA model. The Tamil database contains manually marked syllable boundary information. This is used to derive the syllable segments. For the 1044 syllables chosen as the vocabulary, reference syllable segments are chosen manually. Based on studies in Section 5.7, the length of the documents is restricted to 5 syllables.

The DTW approach is adopted to construct the matrix **W** as described previously. The perplexity of the test set for the syllable-based bi-LSA model using an SVD truncation order of 200 is 24.1. This performance is similar to the text-based syllable level bi-LSA model and better than a text-based syllable level standard bigram model. The perplexities are given in Table 7.5.

We have shown that it is indeed possible to model the large span constraints that exist among the words and the syllables directly from the speech signal. We have also shown that large span constraint models are appropriate for the task.

7.4 SUMMARY

In this chapter the development of a speech-based large span constraint model was explained. Two simple techniques were used to derive the speech-based bi-LSA model.

In the DTW-based approach, the matrix of membership values which is similar to the matrix of counts was developed. For every word segment in the training data, the DTW was performed with each reference word segment in the vocabulary. The distance measure was converted into a membership value which was used to derive the matrix of membership values directly from the speech signal. The performance of this speech-based bi-LSA model was superior to the bigram model, developed both at the word level and the syllable level. The DTW approach was time consuming, as the DTW had to be performed between every word segment in the training data and all the words in the vocabulary. As an alternative, we proposed a template-based approach to develop the speech-based bi-LSA model. The performance of this model was similar to the model derived based on the DTW approach. Smoothing at the word level and document level, and parameter optimisation may further improve the performance of the language model [8]. This method of indirect incorporation of the speech information may be a small step towards using speech level constraints in language models to improve the performance of speech recognition. One limitation in extending the study is the lack of a large speech corpus of the size required for language modelling, segmented in terms of words. A summary of the issues discussed in this chapter is given in Table 7.6. In the next chapter, we extend the use of the large span constraint models to the speaker recognition task.

Table 7.6 Summary of approaches to develop speech-based large span constraint models.

- Large span constraints exist in speech.
- Previous approaches have always modelled constraints from text corpora.
- We proposed two approaches to model the large span constraints directly from the speech signal, using simple matching techniques.
- The performance of these models were better than the standard bigram language model and equivalent text-based large span constraint models.

CHAPTER 8

LARGE SPAN CONSTRAINT MODELS FOR SPEAKER RECOGNITION

8.1 INTRODUCTION

Inconsequently the previous chapters we used the large span constraint model for language modelling. In this chapter we propose to extend the use of large span constraint models to the speaker recognition task. The idea is to see if there exist latent idiolectic characteristics of a speaker that can be modelled in the framework of LSA and whether such a large span constraint model can perform well independently for the speaker recognition task. Alternatively, as in the case of language modelling if the large span constraint speaker model provides sufficient additional constraints (complementary information), it can be used in combination with other speaker recognition approaches to improve the overall performance of the system.

There exist certain traits specific to a speaker that help in easy identification of the speaker among a set of familiar speakers. These include certain disfluencies and mannerisms like stress for certain words, frequent usage of certain phrases or back channel expressions and manner of pronunciation. The focus of this study is identification of a speaker using such idiolectic traits in conversational speech. Every normal conversation by a speaker contains his idiolectic signature. The idiosyncratic patterns in speech are likely to be speaker dependent. These patterns are noticed more in unrestricted conversational speech. They are not so pronounced in read speech or news bulletin type of speech. A large span constraint model is developed to capture the idiolectic signature of each target speaker to be represented in the system. The

similarity of the idiolectic signature in the claimant utterance to that captured by the model is used to hypothesise the target speaker. The technique is demonstrated for the *NIST 2003 extended data task* [118].

In conventional speaker recognition studies, short-time acoustic features are extracted from the speech signal and Gaussian mixture models or neural network models are trained to estimate the distribution of the feature vectors in higher dimensional space [156] [157]. The advantage of such modelling is its simplicity and robustness. These techniques make no effort to capture the higher level knowledge present in speech. To improve the performance of conventional speaker recognition systems, augmenting the scores from the GMM-based system with prosodic and lexical information to identify speakers had been explored [124] [120]. A task similar to speaker recognition based on idiolectic characteristics is that of authorship attribution. The objective of authorship attribution is to establish the authorship of anonymous or doubtful texts based on the idiosyncrasies observed in the previous writings of the authors. The work in this area was reviewed in Section 2.6.

8.2 LATENT SEMANTIC ANALYSIS FOR SPEAKER RECOGNITION

For every conversation of a speaker, the co-occurrences of words or phrases (n-gram) called *terms* in the *conversations*, are sought to be captured in the latent semantic analysis framework. A (*term* $\times$ *conversation*) matrix, $\mathbf{W}$ is constructed with rows representing M terms and columns representing N conversations. Each element $w_{i,j}$ contains the frequency count of the term i in the conversation j. Fig. 8.1 depicts the the block diagram of the proposed system. From the Switchboard corpus based on the NIST 2003 evaluation plan, a set of 4/8/16 conversations belonging to the target speaker are combined and used as the training data for the speaker. The combined conversation is now considered as one conversation for computing the frequency

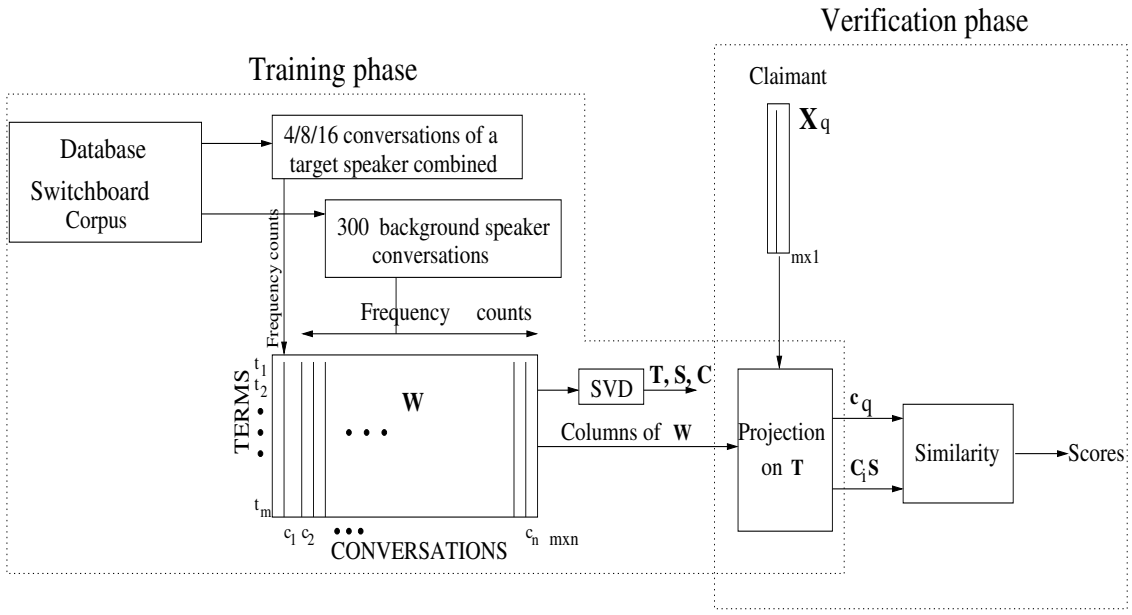

Fig. 8.1 Block diagram of the proposed large span constraint based speaker recognition system.

counts. The frequency counts of the terms in the conversations are used to construct a column of the matrix $\mathbf{W}$ representing the speaker. To facilitate decision making, conversations of background speakers are also used. Background speakers are those speakers who are not part of the training or test set. Conversations from these speakers are used to create the background (reference) model. For each conversation of a background speaker, one column of $\mathbf{W}$ is derived in a manner similar to the target speaker. Thus the matrix W is composed of $N = N_1 + N_2$ columns where, there are N_1 columns corresponding to the target speaker conversations (usually one) and N_2 columns correspond to the background speaker conversations. The matrix $\mathbf{W}$, is decomposed into a product of three matrices using SVD. Only the k largest singular values and their associated vectors are retained such that, $\hat{\mathbf{W}} \approx \mathbf{T}.\mathbf{S}.\mathbf{C}^T$ where, $\mathbf{T}$ and $\mathbf{C}$ are the matrices of singular vectors, and $\mathbf{S}$ is the matrix of singular vectors, similar to that mentioned in Section 5.1.2. Instead of U and V the notation T and C are used

at the matrices are related to the terms and conversations. The matrix $\hat{\mathbf{W}}$ captures the major structural associations in the data. One model ($\mathbf{T}, \mathbf{S}$ and $\mathbf{C}$) is derived for each target speaker. The similarity between a test conversation and the model reveals whether the conversation is similar to that of the target speaker or the background speakers. The use of background speakers also helps in test score normalisation. Term usage (frequency counts) by a speaker is directly used in building the model since this could be an indicator of speaker idiosyncrasies as in authorship attribution studies. Term counts in $\mathbf{W}$ can be weighted by the inverse of the term entropy in the corpus as it emphasises speaker specific terms [132].

If $N_t(s_i)$ is the number of times a term t is spoken by speaker s_i, and $N_t(\mathcal{C})$ is the total number of times the term t has occurred in the corpus $\mathcal{C}$, then the probability that a given token of word t was spoken by speaker s_i is given by

$$P_t(s_i) \quad = \quad \frac{N_t(s_i)}{N_t(\mathcal{C})} \quad . \tag{8.1}$$

Then the entropy of the term t, over all the speakers is given by

$$E_t \quad = \quad -\sum_i P_t(s_i) \log P_t(s_i) \quad . \tag{8.2}$$

During testing a column of terms is constructed from the claimant speaker's conversation ($\mathbf{X}_q$) in a manner similar to the columns of matrix $\mathbf{W}$. The column ($\mathbf{X}_q$) is projected on to the orthonormal basis for the column space of W formed by the column vectors of $\mathbf{T}$. This representation in the reduced R-dimension space is given by $\mathbf{c}_q = \mathbf{X}_q^T \mathbf{TS}$. Likewise the representation of the target speaker/background speaker in the R-dimensional space is obtained by projecting the columns of $\mathbf{W}$ onto the orthonormal basis formed by the columns of $\mathbf{T}$. The rows of $\mathbf{CS}$ represent the projected vectors. The similarity between the claimant speaker and the target speaker or the background speaker is found as the normalised cosine between their respective

projected vectors. If the similarity between the target speaker and the claimant is highest then the claim is accepted, otherwise the claim is rejected. Different similarity measures may be used in arriving at the similarity scores.

8.3 DATABASE FOR IDIOLECTIC SPEAKER RECOGNITION

As part of the NIST 2003 speaker evaluation plan, multiple conversations of a speaker, each of 5 minute duration are available for training. This data is part of the Switchboard corpus phase 2 and 3. The data in this corpus has been organised into different sets containing different combinations of speakers for the evaluation. Set I of the extended data task was considered for our study. This set contains 31 speakers. For each speaker different combinations of 4, 8 or 16 conversations involving him/her are used to build models for that speaker. A total of 265 models are thus built as dictated by the evaluation. The claimant utterances are of 2 minutes duration. Only the text transcriptions pertaining to the conversations are used in the development of the LSA-based model. A total of 3663 tests are conducted against these 265 models. As part of the evaluation auxiliary information was also provided. Some of the information provided are: (1) Text transcription of the entire corpus derived from a speech recognizer (word error rate $\approx 30\%$). (2) Scores obtained from a GMM based speaker recognition system for this evaluation data. (3) Scores from a language modelling based speaker recognition system (LM) and (4) Pitch contour estimates for the entire speech data.

8.4 EXPERIMENTAL EVALUATION

The terms in the matrix $\mathbf{W}$ are n-gram of order 1 to 5, i.e., phrases containing 1 to 5 words. All conversation pertaining to a target speaker (say 4 or 8 or 16) were combined to derive one column ($N_1 = 1$, first column) in the matrix $\mathbf{W}$. Conversations pertaining to 300 background speakers ($N_2 = 300$) are used to derive the other columns

of the matrix $\mathbf{W}$. Thus the matrix $\mathbf{W}$ is of size $M \times 301$, where M is the number of terms. As explained earlier the columns of $\mathbf{W}$ contains the number of times a term has occurred in the conversations pertaining to the speaker which are used for training. SVD is performed on $\mathbf{W}$ retaining only 34 largest singular values and their associated column vectors in the three matrices. The resulting matrices are of the following sizes, $\mathbf{T}_{M\times 34}, \mathbf{S}_{34\times 34}, \mathbf{C}_{301\times 34}$. The order of SVD truncation is derived based on the trade off between maximising noise suppression and minimising reconstruction error. The models were evaluated as described in the Section 8.2. The different similarity measures used in the study are the cosine measure, Pearson correlation, and the Jaccard coefficient [158].

The advantage of the cosine measure is that it is scale invariant and does not depend on the length of the conversation. The cosine measure ($s^{(\mathrm{C})}$) between two vectors $\mathbf{x}_a, \mathbf{x}_b$ is given by

$$s^{(\mathrm{C})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\|_2 \|\mathbf{x}_b\|_2} \tag{8.3}$$

where $\| \bullet \|_2$ denotes the L_2 norm.

Correlation is often used to predict a feature from a highly similar set of objects whose features are known. The normalised Pearson correlation ($s^{(\mathrm{P})}$) is defined as

$$s^{(\mathrm{P})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{2}\left(\frac{(\mathbf{x}_a - \bar{\mathbf{x}}_a)^T(\mathbf{x}_b - \bar{\mathbf{x}}_b)}{\|(\mathbf{x}_a - \bar{\mathbf{x}}_a)\|_2 \|(\mathbf{x}_b - \bar{\mathbf{x}}_b)\|_2} + 1\right) \tag{8.4}$$

where $\bar{\mathbf{x}}_a$ is the average value of $\mathbf{x}$ over all dimensions.

The binary Jaccard coefficient ($s^{(\mathrm{J})}$) measures the degree of overlap between two sets, and is computed as the ratio of shared attributes (words) of $\mathbf{x}_a$ AND $\mathbf{x}_b$ to the number possessed by $\mathbf{x}_a$ OR $\mathbf{x}_b$. For example given the two sets of binary vectors $\mathbf{x}_a = (0, 1, 1, 0)^T$ and $\mathbf{x}_b = (1, 1, 0, 0)^T$, the cardinality of their intersection is 1 and the cardinality of their union is 3, rendering the Jaccard coefficient 1/3. Extending this

to discrete non-negative features the similarity is given by

$$s^{(\mathrm{J})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\|_2^2 + \|\mathbf{x}_b\|_2^2 - \mathbf{x}_a^T \mathbf{x}_b} \tag{8.5}$$

The effectiveness of the lower dimensional representation of the speaker conversations obtained after SVD is compared with the uncompressed raw vector of weighted counts. The scores are obtained using the similarity measures. The similarity between the claimant and the target model (first row of **CS**) results in the score for the target model. Likewise the scores obtained between the claimant and the rest of the rows of **CS** give the background scores. Test utterance normalisation is carried out using the background scores. Using these scores the Detection Cost Function (DCF) is computed to evaluate the performance of the model [118].

8.5 RESULTS

The detection cost for the large span constraint model trained using bigrams is shown in Table 8.1. The list of bigrams used as terms are those which occur at least 4 times in the conversations used for training the model. From the table we see that the DCF for uncompressed representation of the conversations vectors is about the same as that for the representation derived after SVD. Conventionally if a larger number of speakers are used then the background model is better. In such cases the SVD-based representation would perform better compared to the uncompressed representation. Hence results obtained from the SVD based representation of the conversation vectors are reported in the rest of the tables.

As an alternative representation for the target speaker, among the conversations (4/8/16) available for training the target model, one column vector is derived for each of the conversations (i.e., $N_1 = 4/8/16$. Thus the matrix $\mathbf{W}$ would be of size ($M \times 304/308/316$). During testing the similarity score of the claimant with each of

Table 8.1 Detection cost for bigram term types without entropy weighting. The DCF is given for three similarity measures and raw uncompressed vectors.

Term type/ # background Speakers/ # Models per target speaker	Similarity measure used	Detection cost	
		SVD-based representation	Uncompressed representation
Bigrams/ 30 Bg. speakers/ Single model	Correlation	46.95	46.58
	Cosine	46.90	46.51
	Jaccard	46.92	**46.45**
Bigrams/ 30 Bg. speakers/ Multiple models	Correlation	**49.31**	49.51
	Cosine	49.60	49.60
	Jaccard	49.44	49.52
Bigrams/ 300 Bg. speakers/ Single model	Correlation	30.99	**30.01**
	Cosine	32.62	32.19
	Jaccard	31.00	30.17
Bigrams/ 300 Bg. speakers/ Multiple models	Correlation	33.25	34.19
	Cosine	33.75	32.84
	Jaccard	31.88	**31.57**

the 4/8/16 vectors pertaining to the target model is determined. The maximum of these scores is taken as the score for the target speaker. The DCF based on these scores is given as single model per speaker and multiple models per speaker in Table 8.1. Likewise the performance when the terms are unigrams is given in Table 8.2. We see from both the tables that the single model per speaker approach performs better than the multiple models per speaker approach. It is also observed that models using larger number of background speakers (300 instead of 30) perform better.

To determine which type of terms capture the idiolectic constraints better, different types of terms were explored. Unigrams, bigrams, all n-grams of size 5 or less excluding unigrams and all n-grams of size 5 or less including unigrams were used. For these type of terms the DCF of the model is given in Table 8.3. The Detection Error Tradeoff (DET) plots for the system using n-gram of order 1 to 5 and SVD dimension 34 is shown in Fig. 8.2. For the same data set (Set I of NIST 2003 extended data task), scores from the language model based speaker recognition system provided as auxillary data is used to derive the DET curve. This curve is also plotted in Fig. 8.2. These two systems are based on the text transcriptions of the conversations. Another system based on AANN models [159] is trained on the acoustic features derived from the speech signal. Speech segment pertaining to the most frequently occurring 10 words alone are used to train the AANN models. The speaker recognition scores obtained from this system for set I of the data is also plotted in Fig. 8.2. We observe that the Equal Error Rate (EER) for the large span constraint based system and the LM based system are about the same. The DET plots are apart when we used only bigrams or 2 to 5 grams as the terms. This is seen from Fig. 8.3.

We notice that when the large span constraint model scores are combined using the sum rule [160], with any of the other two systems, the performance of the combined

Table 8.2 Detection cost for unigram term types without entropy weighting. The DCF is given for three similarity measures.

Term type/ # background speakers/ # Models per target speaker	Similarity measure used	Detection cost
Unigrams/ 30 Bg. speakers/ Single model	Correlation	25.49
	Cosine	25.30
	Jaccard	**24.00**
Unigrams/ 30 Bg. speakers/ Multiple models	Correlation	25.37
	Cosine	24.93
	Jaccard	**23.70**
Unigrams/ 300 Bg. speakers/ Single model	Correlation	21.35
	Cosine	21.74
	Jaccard	**20.89**
Unigrams/ 300 Bg. speakers/ Multiple models	Correlation	24.28
	Cosine	24.27
	Jaccard	**23.32**

Table 8.3 Detection cost for different term types with and without weighting by entropy for three similarity measures. Single model per speaker and 300 background speakers are used.

Term type	Similarity measure used	Detection cost	
		With entropy weighing	No entropy weighing
Unigrams	Correlation	24.32	21.35
	Cosine	24.27	21.74
	Jaccard	23.32	**20.89**
Bigrams	Correlation	49.31	**30.99**
	Cosine	48.43	32.62
	Jaccard	49.98	31.00
2 to 5 gram	Correlation	**28.12**	32.34
	Cosine	30.39	32.46
	Jaccard	29.11	31.07
1 to 5 gram	Correlation	19.34	19.09
	Cosine	19.68	19.66
	Jaccard	19.02	**19.02**

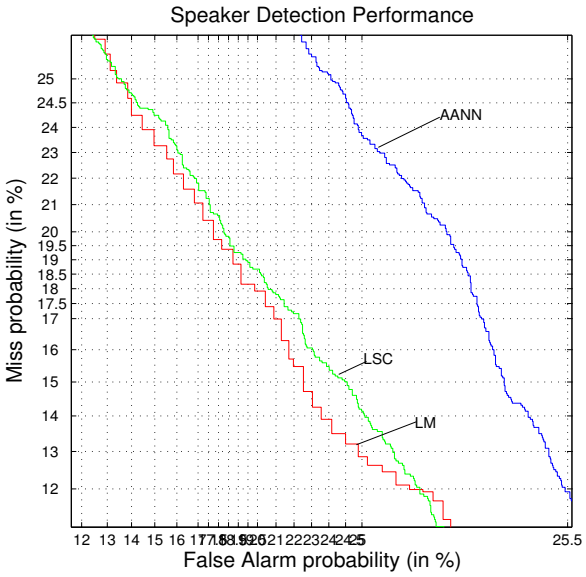

Fig. 8.2 DET plots for the large span constraint (LSC), AANN and Language modelling based systems.

system improves, as seen in Fig. 8.4. The performance is better than any of the individual systems. When the scores from all the three systems are combined then we obtain the best performance. The EER reduces to about 12.0. This suggests that there is significant complementary information that can be derived from the large span constraint based speaker recognition system which would help improve the speaker recognition performance.

8.6 SUMMARY

In this chapter the principles derived from the authorship studies, idiolectic speaker recognition and latent semantic analysis were extended to derive large span constraint models for speaker recognition. A speaker's conversation was represented as a column vector containing the terms (n-gram) spoken. A set of background speakers, not part of the training data, was used to develop a background model. We projected this set of conversation vectors onto a reduced dimensional space using SVD, and this formed our

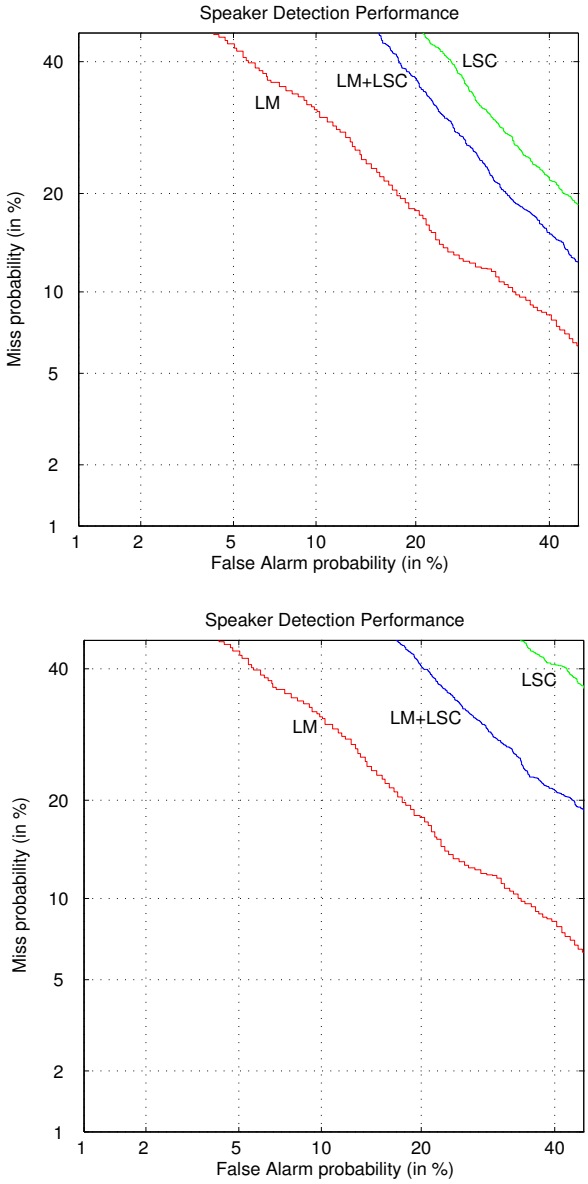

Fig. 8.3 DET plots for the LSC and Language modelling based systems using only bigrams (top) or 2 to 5 grams (bottom) as terms.

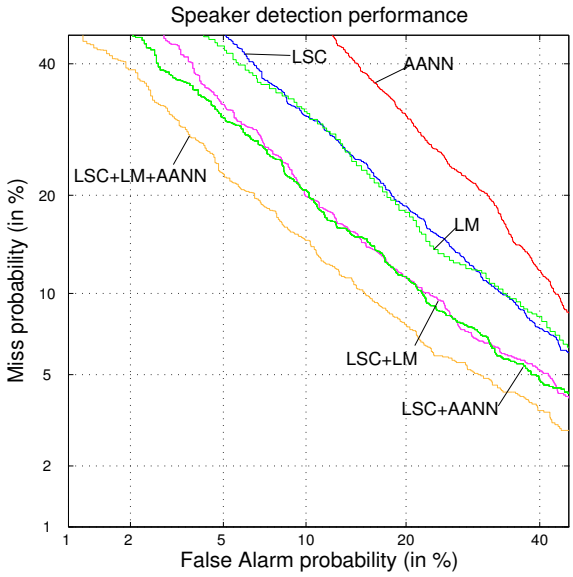

Fig. 8.4 DET plots for combined systems (LSC+LM+AANN).

speaker model. The claimant conversation was also projected on the reduced dimensional space. The similarity between the claimant conversation and the conversation of the target speaker was used to hypothesise the validity of the claim. Different similarity measures and representations of the terms were examined. We found that the model performed as well as other speaker recognition models based on the language modelling principles. This large span constraint model provides complementary information such that when the scores from this model were combined with scores from other systems the performance of the combined system improves. This is despite the fact that the quality of the transcripts are poor ($\simeq$ 30% WER). The issues discussed in this chapter are summarised in Table 8.4. In the next chapter, we summarise the studies carried out in this thesis, list the major contributions of this work and propose directions for future work in this area.

Table 8.4 Summary of approach to model the idiolectic characteristics for speaker recognition.

- Idiolectic characteristics are speaker specific, and are spread over the entire conversation (large spans). They can be used for the speaker recognition task.
- Previous approaches used the n-gram language modelling principles to model the idiosyncrasies.
- The approach proposed in this chapter extended the use of the LSA concept, to develop large span constraint models for the speaker recognition task.
- The performance of the proposed approach was similar to other approaches that are based on the language modelling principles. The model provides complementary information. When the model is combined with other models, the overall performance improves.

CHAPTER 9

SUMMARY AND CONCLUSIONS

9.1 SUMMARY OF THE WORK

In this thesis, we addressed the issues pertaining to the development of large span constraint models for use in speech systems. In most speech systems, the use of higher level knowledge (constraints) improves the performance of the system. Some of the higher level constraints in speech are the syntax and semantics associated with the language spoken. Conventionally, language models are used to capture the syntactic constraints. The large span semantic constraints can be captured using the framework of Latent Semantic Analysis (LSA). These language models have always been derived from large text corpora. For many languages, especially non-literary languages, such text corpora do not exist. For such languages, it is easy to record speech data in large volumes with minimal effort. In this work we had described approaches to derive the large span constraints directly from the speech signal using the framework of latent semantic analysis.

With the speech recognizer as a possible application, language models capturing large span constraints were developed. The framework of latent semantic analysis requires the construction of a matrix ($\mathbf{W}$) of co-occurrences of words in documents (here spoken documents) to capture the semantic constraints. This matrix was derived directly from the speech signal using simple matching techniques like Dynamic Time Warping (DTW) and template matching. In the dynamic time warping approach, the speech segment corresponding to a word in the training document was matched with each reference word segment corresponding to the words in the vocabulary. Based on

the dissimilarity measure, a set of closest matching words were found. The values in the corresponding locations of these words in **W** were incremented. This procedure was carried out for all the words in the training document, and for all the documents in the corpus. This matrix, derived from the speech signal is similar to the matrix of co-occurrences derived from a text corpus. It formed the basis for the LSA-based model. The model captured the large span constraints that exist among the words. We showed that when such a model is combined with a bigram language model, the performance of the combined bi-LSA model is better than the bigram model.

In the above approach, DTW has to be performed between each word segment in the training corpus and each reference word segment in the vocabulary. For a medium sized vocabulary and a large training corpus, the approach would be time consuming. Hence as an alternative, a template-based approach was proposed. In this approach a word is represented by a fixed dimensional pattern vector derived from the speech signal by concatenating the feature vectors corresponding to selected frames of the word segment. This large dimensional vector was compressed to a smaller dimension using Autoassociative Neural Network (AANN). This reduced dimensional pattern vector represents a word segment. The similarity between a reduced dimensional pattern vector in the training data and reference word pattern vector was used to obtain the closest matching words. The rest of the procedure in constructing **W** and the bi-LSA model is same as that in the DTW approach. We showed that the performance of the the speech-based large span constraint model using the template-based approach is similar to the performance of the model derived using the DTW approach.

The large span constraint models were also derived at the syllable level using both the text and speech corpus. It was shown that there exist large span constraints at the syllable level also. The syllable level large span constraint model was combined with

the syllable-based bigram language model to give the bi-LSA model. The performance of the speech-based bi-LSA model was shown to be better than the bigram language model both at the word and syllable level. The performance was better than the equivalent text-based large span constraint models, at the word level and about the same at the syllable level.

One of the main issues while developing a speech-based large span constraint model is the possibility of occurrence of errors in the recognition/matching stage. In speech recognition the common errors that occur are the insertion, deletion and substitution errors. Usually a combination of all the three errors occur. The effect of these speech recognition errors were simulated on the word-based matrix $\mathbf{W}$ derived from the text corpora. It was found that the performance of the large span constraint language model was not significantly affected for errors (either individual or in combination) in the transcription up to 20%.

Language models are predominantly used in speech recognizers. The first step in the development of a speech recognizer is to identify the subword unit to model. From an information theoretic perspective it was shown that syllables have a higher redundancy when compared to conventionally used subword units like phonemes, biphones and triphones. A unit with higher redundancy captures the local constraints better, and is more suitable for recovery from errors. Thus syllables are appropriate as subword units for speech recognition. Their use is also appropriate from the speech production and perception point of view. To develop syllable models for the speech recognizer, statistical characteristics of the syllables that occur in the IITM speech corpus were studied. The studies suggested that modelling a small subset comprising of the most frequently occurring syllables in the syllable vocabulary is likely to cover most of the syllable occurrences (about 88%) in the corpus. The studies were used to

arrive at some of the parameters useful for developing syllable models using the HMM framework.

A syllable-based recognizer was developed for two Indian languages, Tamil and Telugu. To improve the recognition rate of the syllables, neural network-based pre-classifiers were developed to classify the syllables into groups (equivalent classes). A conventional HMM-based syllable recognizer was then used to hypothesize the syllable within the group. This reduced the search space and improved the recognition rate of the syllables.

In conversational speech, certain idiolectic characteristics specific to a speaker can be identified. These idiolectic characteristics are spread over large spans (entire conversation). It is possible to use these characteristics to identify a speaker. The large span constraint model was extended to the speaker recognition task. The latent idiolectic characteristics of a speaker were captured using this model. This large span constraint speaker recognition system performed as well as other speaker recognition systems that are based on language modelling principles. The system also provided complementary information. Hence when the scores from the large span constraint based speaker recognition system were combined with the scores of other speaker recognition systems, the performance of the combined system improved.

9.2 MAJOR CONTRIBUTIONS OF THE WORK

1. Large span constraint language models were derived directly from the speech signal for the first time.
2. Approaches based on simple matching techniques were proposed to derive the speech-based large span constraint language model.
3. It was shown that large span constraints exist at the syllable level also. An

approach was proposed to construct a large span constraint model at the syllable level.

4. The performance of the speech-based bi-LSA model was shown to be better than the bigram language model both at the word and syllable level. The performance was better than the equivalent text-based large span constraint models, at the word level and about the same at the syllable level.

5. The latent idiolectic characteristics of a speaker was captured using large span constraint models. This approach was extended the use of the LSA concept for the speaker recognition task.

6. A support vector machine-based preclassifier was proposed to reduce the search space in the speech recognizer and to improve its performance.

7. From an information theoretic perspective, it was shown that the syllable was an appropriate subword unit for speech recognition.

9.3 DIRECTIONS FOR FUTURE WORK

1. Build a complete syllable-based speech recognition system: In this work we develop a syllable recognizer. The vocabulary of the recognizer is a small subset of the syllables in the language. Methods of developing acoustic models for all the syllables in the language needs to be addressed. As most of the syllables are infrequent, it may involve tying the states and mixtures of the HMM model. Decision tree clustering approach to tie the states and pool the data may be adopted. However the decision making questions for the procedure are not readily available for any Indian language. A larger database for training is also required. It should be preferably segmented in terms of syllables.

2. Derive the bigram language models directly from the speech data: In the ap-

proach adopted to develop the bi-LSA model, the bigram model was developed from a text corpus. Approaches to deriving the bigram models directly from the speech signal need to be developed. This would result in a better integration of the bigram model with the speech-based LSA model.

3. Incorporate the speech-based bi-LSA model into the speech recognizer: Language models are primarily used in a speech systems. The speech-based bi-LSA model has to be integrated into a speech recognizer. As a complete speech recognizer is not available for any Indian language, this integration is not yet carried out. As the language model is of large span, one approach to complete the integration is as follows. Word lattices along with the acoustic and language model scores can be derived from the speech recognizer. The lattice can be rescored with the large span language modelling probabilities. The best scoring 1-best hypothesis can then be derived.

BIBLIOGRAPHY

[1] Joshua T. Goodman, "A bit of progress in language modeling," *Comput. Speech Lang.*, vol. 15, pp. 403–434, Oct. 2001.

[2] John Makhoul, Francis Kubala, Timothy Leek, Daben Liu, Long Nguyen, Richard Schwartz, and Amit Srivastava, "Speech and language technologies for audio indexing and retrieval," *Proc. IEEE*, vol. 88, pp. 1338–1353, Aug. 2000.

[3] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien, "Using linear algebra for intelligent information retrieval," Tech. Rep. UT-CS-94-270, CSE Dept., Univ. of Tennessee, Dec. 1994.

[4] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Information Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[5] Thomas K. Landauer, Peter W. Foltz, and Darrel Laham, "An introduction to latent semantic analysis," *Discourse Process*, vol. 25, pp. 259–284, 1998.

[6] Jerome R. Bellegarda, "A latent semantic analysis framework for large-span language modeling," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Rhodes, Greece), pp. 1451–1454, Sept. 1997.

[7] N. Coccaro and D. Jurafsky, "Toward better integration of semantic predictors in statistical language modeling," in *Proc. Int. Conf. Spoken Language Processing*, (Sydney, Australia), pp. 2403–2406, Dec. 1998.

[8] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proc. IEEE*, vol. 88, pp. 1279–1296, Aug. 2000.

[9] Languages in India 1991, "http://www.refiff.com/news/2004/dec/13india.htm."

[10] F. Jelinek, *Statistical Methods for Speech Recognition.* Cambridge MA: MIT Press, 1997.

[11] Lawrence. R. Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition.* New Jersy: PTR Prentice Hall Inc, 1993.

[12] Jean-Luc Gauvain and Lori Lamel, "Large-vocabulary continuous speech recognition: Advances and applications," *Proc. IEEE*, vol. 88, pp. 1181–1200, Aug. 2000.

[13] A. Tsopanoglou and N. Fakotakis, "Selection of the most effective set of subword units for a HMM-based speech recognition system," in *Proc. Eur. Conf. Speech Commun. Technol. 97*, vol. 3, (Rhodes, Greece), pp. 1231–1234, Sept. 1997.

[14] Thilo Pfau, Manfred Beham, W. Reichl, and Gnther Ruske, "Creating large subword units for speech recognition," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 1191–1194, 1997.

[15] Oh-Wook Kwon, "Performance of lvcsr with morpheme-based and syllable-based recognition units," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 3, (Istanbul, Turkey), pp. 1567–1570, June 2000.

[16] P. Geutner, "Using morphology towards better large vocabulary speech recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Michigan, USA), pp. 445–448, May 1995.

[17] Laura Mayfield Tomokiyo and Klaus Ries , "An automatic method for learning Japanese lexicon for recognition of spontaneous speech," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Seattle, USA), pp. 305–308, May 1998.

[18] K. C. Yang, T. -H, Ho, L. -F. Chien, and L. -S Lee, "Statisticsr-based segment pattern lexicon - a new direction for Chinese language modelling," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Seattle, USA), pp. 169–172, May 1998.

[19] Toshiaki Fukada, Michiel Bacchiani, Kuldip K. Paliwal, and Yoshinori Sagisaka, "Speech recognition based on acoustically derived segment units," in *Proc. Int. Conf. Spoken Language Processing*, vol. 2, (Philadelphia, USA), pp. 1077–1080, Oct. 1996.

[20] H. Gish and K.Ng, "A segmental speech model with applications to word spotting," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 2, (Minnesota, USA), pp. 447–450, Apr. 1993.

[21] Sabine Deligne and Frederic, "Inference of variable-length acoustic units for continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 3, (Munich, Germany), Apr. 1997.

[22] Sabine Deligne, F. Yvon, and Frédéric Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Madrid, Spain), pp. 2243–2246, Sept. 1995.

[23] Frédéric Bimbot, Roberto Pieraccini, Esther Levin, and Bishnu Atal , "Variable-length sequence modelling: Multigrams," *IEEE Signal Processing Lett.*, vol. 2, pp. 111–113, June 1995.

[24] Sabine Deligne and Frédéric Bimbot, "Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Michigan, USA), pp. 169–172, May 1995.

[25] S. V. Gangashetty, A. Nayeemulla Khan, S. R. Mahadeva Prasanna, and B. Yegnanarayana, "Neural network models for preprocessing and discriminating utterances of consonant-vowel units," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 1, (Honolulu, Hawaii, USA), pp. 613–618, May 2002.

[26] Steven Greenberg, "Speaking in shorthand - A Syllable-centric perspective for understanding pronunciation variation," *Speech Comm.*, vol. 29, pp. 159–176, Nov. 1999.

[27] P. Eswar, S. K. Gupta, C. Chandra Sekhar, B. Yegnanarayana, and K. Nagamma Reddy, "An acoustic-phonetic expert for analysis and processing of continuous speech in Hindi," in *Proc. European Conf. Speech Technology*, (Edinburgh, UK), pp. 369–372, Sept. 1987.

[28] S. V. Gangashetty, K. Sreenivasa Rao, A. Nayeemulla Khan, C. Chandra Sekhar, and B. Yegnanarayana, "Combining evidence from multiple modular networks for recognition of consonant-vowel units of speech," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 1, (Portland, Oregon, USA), pp. 686–691, July 2003.

[29] Alfred Hauenstein, "Using syllables in a hybrid HMM-ANN recognition system," in *Proc. Eur. Conf. Speech Commun. Technol. 97*, vol. 3, (Rhodes, Greece), pp. 1203–1206, Sept. 1997.

[30] H. Bourlard, "Towards increasing speech recognition error rates," *Speech Comm.*, vol. 18, pp. 205–231, May 1996.

[31] Osamu Fujimura, "Syllable as a unit of speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23(1), pp. 82–87, Feb. 1975.

[32] Rhys James Jones, Simon Downey, and John S. Mason, "Continuous speech recognition using syllables," in *Proc. Eur. Conf. Speech Commun. Technol. 97*, (Rhodes, Greece), pp. 1171–1174, Sept. 1997.

[33] R. A. Cole, M. Noel, T. Lander, and T. Durham, "New telephone speech corpora at CLSU," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Madrid, Spain), pp. 821–824, Sept. 1995.

[34] Su-Lin Wu, Brain E. D. Kingsbury, Nelson Morgan, and Steven Greenberg, "Performance improvements through combining phone and syllable-scale information in automatic speech recognition," in *Proc. Int. Conf. Spoken Language Processing*, (Sydney, Australia), Dec. 1998.

[35] Issam Bazzi and James Glass, "Heterogenous lexical units for automatic speech recognition: Preliminary investigations," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, (Istanbul), pp. 1257–1260, 2000.

[36] Sadaoki Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 52–59, Feb. 1986.

[37] C. Chandra Sekhar, *Neural network models for recognition of Stop Consonant-Vowel (SCV) segments in continuous speech*. PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras, Apr. 1996.

[38] O. Ghitza and M. M. Sondhi, "Hidden Markov models with templates as non-stationary states: An application to speech recognition," *Comput. Speech Lang.*, vol. 7, pp. 101–119, Apr. 1993.

[39] M. Fanty, R. Cole, and K. Roginski, *Advances in neural infromation processing systems 4*, ch. English alphabet recognition with telephone speech. San Mateo, CA: Morgan Kaufmann, 1992.

[40] A. Nayeemulla Khan and B. Yegnanarayana, "Vowel onset ponit based variable frame rate analysis for speech recognition," in *Proc. IEEE Int. Conf. Intelligent Sensing and Information Processing*, (Chennai, India), pp. 392–394, Jan. 2005.

[41] Zhihong Hu, Johan Schalkwky, Etienne Barnard, and Ronald Cole, "Speech recognition using syllable-like units," in *Proc. Int. Conf. Spoken Language Processing*, vol. 2, (Philadelphia, PA, USA), pp. 1117–1120, Oct. 1996.

[42] J. Y. Siva Rama Krishna Rao, C. Chandra Sekhar, and B. Yegnanarayana, "Neural networks based approach for detection of vowel onset points," in *Proc. Int. Conf. Advances in Pattern Recognition and Digital Techniques (ISI calcutta, India)*, pp. 316–320, Dec. 1999.

[43] S. V. Gangashetty, C. Chandra Sekhar, and B. Yegnanarayana, "Detection of vowel onset points in continuous speech using autoassociative neural network models," in *Int. Conf. Spoken Language Processing (INTERSPEECH 2004 - ICSLP)*, vol. 2, (Jeju Island, Korea), pp. 1081–1084, Oct. 2004.

[44] S. V. Gangashetty, C. Chandra Sekhar, and B. Yegnanarayana, "Dimension reduction using autoassociative neural network models for recognition of consonant-vowel units of speech," in *Fifth Int. Conf. Advances in Pattern Recognition*, (ISI Calcutta, India), pp. 156–159, Dec. 2003.

[45] J. Y. Siva Rama Krishna Rao, "Recognition of Consonant-vowel (CV) utterances using modular neural network models," MS thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras, May. 2000.

[46] C. Chandra Sekhar and B. Yegnanarayana, "A Constraint satisfaction model for recognition of Stop Consonant-Vowel (SCV) utterances," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 472–480, Oct. 2002.

[47] A. Nayeemulla Khan, "Recognition of syllable like units in Indian languages," in *Proc. Int. Conf. Natural Language Processing*, (Mysore India), pp. 135–141, Dec. 2003.

[48] S. V. Gangashetty, *Neural network models for recognition of consonant-vowel units of speech in multiple languages.* PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras, Feb. 2005.

[49] Nelson Morgan and Hervé Bourlard, "Continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 12, pp. 24–42, May 1995.

[50] Steve Young, "Large vocabulary continuous speech recognition: A review," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, (Snowbird, Utah), pp. 3–28, Dec. 1995.

[51] Steve Renals and Nelson Morgan, "Connectionist probability estimation in HMM speech recognition," Tech. Rep. TR-92-081, ICSI, Berkely, California, Dec. 1992.

[52] Joseph Picone, "Continuous speech recognition using hidden Markov models," in *IEEE ASSP Magazine*, pp. 26–41, July 1990.

[53] Steve Young, "Statistical modelling in continuous speech recognition (CSR)," in *Proc. Intl. Conf. on Uncertainty in Artificial Intelligence*, (Seattle, WA, USA), Aug. 2001.

[54] J. S. Bridle and M. D. Brown, "Connected word recognition using whole word templates," in *Proc. Inst. Acoust. U.K*, pp. 25–28, 1979.

[55] Steve J. Young, "A review of large-vocabulary continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 13, pp. 45–57, Sept. 1996.

[56] Wendy Holmes and Mark Huckvale, "Why have HMMs been so successful for automatic speech recognition and how might might they be improved," in *Speech, Hearing and Language, UCL Work in Progress*, vol. 8, (University College London), pp. 207–219, 1994.

[57] S. E. Levinson, "Structural methods in automatic speech recognition," *Proc. IEEE*, vol. 73, pp. 1625–1650, Nov. 1985.

[58] J. N. Holmes, "Use of phonetic knowledge when designing and training stochastic models for speech recognition," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Genova, Italy), pp. 1257–1260, Sept. 1991.

[59] M. J. F. Gales and S. J. Young, "Segmental hidden Markov models," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 3, (Berlin, Germany), pp. 1579–1582, Sept. 1993.

[60] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1641–1648, Nov. 1989.

[61] C. H. Lee, L. R. Rabiner, R. Pieraccini, and J. G. Wilpon, "Acoustic modeling for large vocabulary speech recognition," *Comput. Speech Lang.*, vol. 4, pp. 127–165, Jan. 1990.

[62] K. F. Lee, "Context-dependent phonetic hidden Markov models for speaker independent continuous speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 599–609, Apr. 1990.

[63] John Sirigos, Nikos Fakotakis, and George Kokkinakis, "A hybrid ANN/HMM syllable recognition module based on vowel spotting," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 3, (Budapest, Hungary), pp. 1119–1122, Sept. 1999.

[64] P. Mermelstien, "Automatic segmentation of speech into syllablic units," *J. Acoust. Soc. Amer.*, vol. 58, pp. 880–883, Oct. 1975.

[65] V. Kamakshi Prasad, *Segmentation and recognition of continuous speech.* PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras, Jan. 2002.

[66] S. R. Mahadeva Prasanna, *Event based analysis of speech.* PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras, Jan. 2004.

[67] S. R. Mahadeva Prasanna, S. V. Gangashetty, and B. Yegnanarayana, "Significance of vowel onset point for speech analysis," in *Proc. Sixth Biennial Conf. Signal Processing and Communication 2001*, (Bangalore, India), pp. 81–88, July 2001.

[68] T. Waardenburg, J. A. du Preez, and M. W. Coetzer, "The automatic recognition of stop consonants using hidden Markov models," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (California, USA), pp. 585–588, Mar. 1992.

[69] L. Deng and K. Erler, "Microstructural speech units and their HMM representation for discrete utterance speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, pp. 193–196, May 1991.

[70] L. Deng, M. Lennig, and P. Mermelstein, "Modeling microsegments of stop consonants in a hidden Markov model based word recogniser," *J. Acoust. Soc. Amer.*, vol. 87, pp. 2738–2747, June 1990.

[71] Tanee Demechai and Kimmo Mäkeläinen, "Recongition of syllables in a tone language," *Speech Comm.*, vol. 33, pp. 241–254, Feb. 2001.

[72] Hsin-min Wang, "Experiments in syllable-based retrieval of broadcast news speech in Mandarin Chinese," *Speech Comm.*, vol. 32, pp. 49–60, Sept. 2000.

[73] Hsiao-Wen Hon, Baosheng Yuan, Yen-Lu Chow, Shankar Narayan, and Kai-Fu Lee, "Towards large vocabulary Mandarian Chinese speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Adelaide, Australia), pp. 545–548, Apr. 1994.

[74] Aravind Ganapathiraju, Jonathan Hamaker, Joseph Picone, Mark Ordowski, and Gorge R. Dodington, "Syllable-based large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 358–366, May 2001.

[75] L. R. Bhal, J. K. Baker, F. Jelinek, and R. L. Mercer, "Perplexity-a measure of the difficulty of speech recognition tasks," *J. Acoust. Soc. Amer.*, vol. 62, p. S63, 1977. Suppl. no. 1.

[76] Ronald Rosenfeld, "Two decades of statistical language modeling: Where do we go from here," *Proc. IEEE*, vol. 88, pp. 1270–1278, Aug. 2000.

[77] Ronald Rosenfeld, *Adaptive statistical language modeling: A maximum entropy approach.* PhD thesis, School of Computer Science, Carnegie Mellon University, Apr. 1994.

[78] I. Guyon and F. Pereira, "Design of a linguistic postprocessor using variable memory length Markov models," in *Proc. 3rd ICDAR*, (Montreal, Canada), pp. 454–457, Aug. 1995.

[79] T. Niesler and P. Woodland, "Variable-length category n-gram language models," *Comput. Speech Lang.*, vol. 13, pp. 99–124, Jan. 1999.

[80] M. -H. Siu and M. Ostendorf, "Variable n-gram and extensions for conversational speech language modeling," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 63–75, Jan. 2000.

[81] Antonio Bonafonte and José B. Mariño, "Language modeling using x-grams," in *Proc. Int. Conf. Spoken Language Processing*, vol. 1, pp. 394–397, Oct. 1996.

[82] X. Huang, F. Alleva, H. -W. Hon, M. -Y, Hwang, K. -F. Lee, and R. Rosenfeld, "The SPIHINX-II speech recognition system: An overview," *Comput. Speech Lang.*, vol. 7, pp. 137–148, Apr. 1993.

[83] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modeling," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Berlin, Germany), pp. 973–976, Sept. 1993.

[84] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[85] P. J. Price, "Evaluation of spoken language systems: The ATIS domain," in *Proc. DARPA Speech and Natural Language Workshop*, June 1990.

[86] W. H. Ward, "The CMU air travel information service: Understanding spontaneous speech," in *Proc. DARPA Speech and Natural Language Workshop*, (Pennsylvania, USA), pp. 127–129, June 1990.

[87] L. R. Bhal, P. F. Brown, P. V. de Souza, and R. L. Mercer, "A tree based statistical language model for natural language speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1001–1008, July 1989.

[88] Rukmini M. Iyer and Mari Ostendorf, "Modeling long distance dependence in language: Topic mixtures versus dynamic chache models," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 30–39, Jan. 1999.

[89] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," *Comput. Speech Lang.*, vol. 10, pp. 187–228, July 1996.

[90] Ronald Rosenfeld, S. F. Chen, and X. Zhu, "Whole-sentence exponential language models a vehicle for linguistic statistical integration," *Comput. Speech Lang.*, vol. 15, pp. 55–73, Jan. 2001.

[91] S. Chen and R. Rosenfeld, "Efficient sampling and feature extraction in whole sentence maximum entropy language models," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Arizona, USA), pp. 549–552, Mar. 1999.

[92] C. Chelba and F. Jelinek, "Recognition performance of a structured language model," in *Proc. 6th Eur. Conf. Speech Commun. Technol.*, vol. 4, (Budapest, Hungary), pp. 1567–1570, Sept. 1999.

[93] F. Jelinek and C. Chelba, "Putting language into language modeling," in *Proc. 6th Eur. Conf. Speech Commun. Technol.*, vol. 1, (Budapest, Hungary), pp. KN1–KN5, Sept. 1999.

[94] C. Chelba and F. Jelinek, "Exploiting syntactic structure for language modeling," in *Proc. 36th Conf. on ACL*, vol. 1, (Qubec, Canada), pp. 225–231, 1998.

[95] E. Charnaik, "Immediate-head parsing for language models," in *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics, 2001*, (Toulouse, France), pp. 124–131, July 2001.

[96] X. J. Zhu, S. F. Chen, and R. Rosenfeld, "Linguistic features for whole sentence maximum entropy language models," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 4, (Budapest, Hungary), pp. 1807–1810, Sept. 1999.

[97] Salim Roukos , *Survey of the state of the art in human language technology*, ch. Language reprentation. Cambridge, U.K: Cambridge Univ. press, 1997.

[98] Thomas K. Landauer and Susan T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, pp. 211–240, 1997.

[99] D. Laham, "Latent semantic analysis approaches to categorization," in *Proc. of the 19th Annual Conf. of the Cognitive Science Society* (M. G. Shafto and P. Langley, eds.), (Hillsdale, NJ, USA), p. 979, Lawrence Erlbaum Associates, Inc, 1997.

[100] T. K. Landauer, *The psychology of learning and motivation*, vol. 41, ch. On the computational basis of learning and cognition: Arguments from LSA, pp. 43–84. 2002.

[101] Thomas Hofmann, "Probabilistic latent semantic analysis," in *Proc. ACM-SIGIR Intl. Conf. on Research and Development in Information Retrieval*, (Berkeley, California, USA), pp. 50–57, Aug. 1999.

[102] Roger E. Story, "An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model," *Information Processing and Management*, vol. 32, no. 3, pp. 329–344, 1996.

[103] Gorge W. Furnas, Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, Richard A. Harshman, Lyan A. Streeter, and Karen E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," in *Proc. Intl. ACM SIGIR Conf. on research and development in information retrieval*, (Grenoble, France), pp. 465–480, 1988.

[104] D. Hull, "Improving text retrieval for the routing problem using latent semantic indexing," in *Proc. 17th ACM/SIGR Conf. on Research and Development in Information Retrieval*, (Dublin, Ireland), pp. 282–290, 1994.

[105] P. Husbands, H. Simon, and C. Ding, "On the use of singular value decomposition for text retrieval," in *Proc. of SIAM Computational Information Retrieval Workshop*, (Philadelphia, USA), pp. 145–156, 2001.

[106] Mikko Kurimo, "Fast latent semantic indexing of spoken documents by using self-organizing maps," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 4, (Istanbul, Turkey), pp. 2425–2428, June 2000.

[107] Samuel Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 1, (Piscataway, NJ), pp. 413–418, May 1998.

[108] Hinrich. Schütze, "Dimensions of meaning," in *Proc. ACM/IEEE Conf. on Supercomputing*, (Minnesota, USA), pp. 787–796, 1992.

[109] April Kontostathis and William M. Pottenger, "Detecting patterns in the LSI term-term matrix," Tech. Rep. LU-CSE-02-010, Dept. CSE. Lehigh Univ., 2002.

[110] April Kontostathis and William M. Pottenger, "A mathematical view of latent semantic indexing: Tracing term co-occurrences," Tech. Rep. LU-CSE-02-006, Dept. CSE. Lehigh Univ., 2002.

[111] April Kontostathis, William M. Pottenger, "Improving retrieval performance with positive and negative equivalence classes of terms," Tech. Rep. LU-CSE-02-009, Dept. CSE. Lehigh Univ., 2002.

[112] April Kontostathis, William M. Pottenger, and Brian D. Davison, *Data Mining: Foundations, Methods, and Applications*, ch. Identification of critical values in latent semantic indexing. Springer-Verlag, 2005.

[113] J. R. Bellegarda, "Exploiting both local and global constraints for multi-span statistical language modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 2, (Seattle, WA), pp. 677–680, May 1998.

[114] Dharmendra Kanejiya, Arun Kumar, and Surendra Prasad, "Statistical language modeling using syntactically enhanced LSA," in *Workshop on spoken language processing*, (TIFR, Mumbai, India), pp. 93–100, Jan. 2003.

[115] Jerome R. Bellegarda and Kim E. A. Silverman, "Toward unconstrained command and control: Data-driven semantic inference," in *Proc. Int. Conf. Spoken Language Processing*, vol. 1, (Beijing, China), pp. 258–261, Oct. 2000.

[116] Rong Zhang and Alexander I. Rudnicky, "Improve latent semantic analysis based language model by integrating multiple level knowledge," in *Proc. ICSLP 02*, (Denver, Colorado), pp. 893–896, Sept. 1996.

[117] Jen Tzung Chein, Meng-Sung Wu, and Hua-Jui Peng, "Latent semantic language modeling and smoothing," *Computational Linguistics and Chinese Language Processing*, vol. 9, pp. 29–44, Aug. 2004.

[118] http://www.nist.gov/speech/tests/spk/2003/, "The 2003 NIST Speaker Recognition Evaluation."

[119] Aaron Rosenberg, "Automatic speaker verification: A review," *Proc. IEEE*, vol. 64, pp. 475–487, Apr. 1976.

[120] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Amer.*, vol. 52, no. 6, pp. 1687–1697, 1972.

[121] John D. Markel, "Long term feature averaging for speaker recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, pp. 330–337, Aug. 1977.

[122] M. Mathew, B. Yegnanarayana, and R.Sundar, "A neural network-based text-dependent speaker verification system using suprasegmental features," in *Proc. Eur. Conf. Speech Commun. Technol.*, (Budapest, Hungary), pp. 95–998, Sept. 1999.

[123] G. R. Doddington, "Speaker recognition - identifying people by their voices," *Proc. IEEE*, vol. 73, pp. 1651–1664, Nov. 1985.

[124] F. Weber, L. Manganaro, B. Peskin, and E. Shriberg, "Using prosodic and lexical information for speaker identification," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Orlando, Florida, USA), pp. 141–144, May 2002.

[125] M. B. Malyutov, "Authorship attribution of texts: a review," in *Proc. of the program Information transfer*, (University of Bielefeld, Germany), 2002.

[126] F. Mosteller and D. Wallace, *Inference and Disputed Authorship.* Addison-Wesley, Reading, Mass., 1964.

[127] Glenn Fung, "The disputed federalist papers: SVM and feature selection via concave minimization," in *Proc. 2003 Conf. Diversity in Computing, TAPIA'03,*, (Atlanta, Georgia), pp. 42–46, Oct. 2003.

[128] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Automatic authorship attribution," in *Proc. nineth Conf. European Chap. Assoc. Computational Linguistics*, (Bergen, Norway), pp. 158–164, Jun. 1999.

[129] F. Peng, D. Schuurmans, V. Keselj, and S. Wang, "Language independent authorship attribution using character level language models," in *10th Conf. European Chapter of the Association for Computational Linguistics (EACL-03)*, (Budapest, Hungary), Apr. 2003.

[130] Susan T. Dumais and Jakob Nielsen, "Automating the assignment of submitted manuscripts to reviewers," in *Proc. of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, (Copenhagen, Denmark), pp. 233–244, June 1992.

[131] David Yarowsky and Radu Florian, "Taking the load off the conference chairs: Towards a digital paper-routing assistant," in *Proc. of the 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very-Large Corpora*, (University of Maryland, College Park, MD, USA), June 1999.

[132] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 4, (Scandinavia), pp. 2521–2524, Sept. 2001.

[133] A. Nayeemulla Khan, S. V. Gangashetty, and S. Rajendran, "Speech database for Indian languages- A preliminary study," in *Proc. Int. Conf. Natural Language Processing*, (NCST, Mumbai, India), pp. 295–301, Dec. 2002.

[134] R. Carlson, K. Elenius, B. Granstrom, and S. Hannicutt, "Phonetic properties of the basic vocabulary of five european languages: Implications for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 4, (Tokyo, Japan), pp. 2763–2766, Apr. 1986.

[135] David W. Shipman and Victor W. Zue, "Properties of large lexicons: Implications for advanced word recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Paris, France), pp. 546–549, May 1982.

[136] K. Nagamma Reddy, *Speech Technology Issues and Implications in Indian Languages*, ch. Relevance of duration for speech system in Telugu, pp. 185–213. Trivandrum, India: Int. School of Dravidian Linguistics, 2000.

[137] J. Sakuntala Sharma and K. Nagamma Reddy, *Speech Technology Issues and Implications in Indian Languages*, ch. Statistical patterns of phonemes and consonant sequences in some Indian languages: Their relevance to speech technology, pp. 61–76. Trivandrum, India: Int. School of Dravidian Linguistics, 2000.

[138] K. Prakash Rao, Akshar Bharati, Rajeev Sangal, and S. M. Bendre, "Basic statistical analysis of corpus and cross comparision among corpora," in *Proc. Int. Conf. Natural Language Processing*, (NCST, Mumbai, India), pp. 121–129, Dec. 2002.

[139] Tanja Schultz, "Globalphone: A multilingual speech and text database developed at karlsruhe university," in *Proc. Int. Conf. Spoken Language Processing*, vol. 1, (Denver, CO), pp. 345–349, Sept. 2002.

[140] Marie-José Caraty and Claude Montacié, "Teraspeech 2000: A 10,000 speakers database," in *Proc. Int. Conf. Spoken Language Processing*, vol. 3, (Beijing, China), pp. 973–976, Oct. 2000.

[141] A. Eskenazi, C. Hogan, J. Allen, and R. Frederking, "Issues in database design: Recording and processing speech from new populations," in *Proc. Int. Conf. on Lang. Resources and Evaluation*, vol. 2, (Granada, Spain), pp. 1289–1293, May 1998.

[142] K. Sreenivasa Rao and B.Yegnanarayana, "Modeling syllable duration in indian languages using neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Quebec, Canada), pp. 109–112, May 2004.

[143] Avinash Chopde, "ITRANS Indian Language Transliteration package Version 5.2 source http://www.aczone.com/itrans/."

[144] William M. Fischer, George R. Doddington, and Kathleen M. Goudie-Marshal, "The DARPA speech recognition research database: specifications and status," in *Proc. DARPA workshop on Speech Recognition*, pp. 93–99, Feb. 1986.

[145] G. K. Zipf, *Human Behaviour and Principle of Least Effort.* Cambridge, UK: Addison-Wesley Press Inc, 1949.

[146] A. Nayeemulla Khan, S. V. Gangashetty, and B. Yegnanarayana, "Neural network preprocessor for recognition of syllables," in *Proc. IEEE Int. Conf. Intelligent Sensing and Information Processing*, (Chennai, India), pp. 171–175, Jan. 2004.

[147] Alex Waibel, Petra Geutner, Laura Mayfield Tomokiyo, Tanja Schlutz, and Monika Woszczyna, "Multilinguality in speech and spoken language systems," *Proc. IEEE*, vol. 88, pp. 1297–1313, Aug. 2000.

[148] Ronald Rosenfeld, "The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation," in *Proc. Spoken Language Systems Technology Workshop*, Mar. 1994.

[149] Takashi Otsuki, Shozo Makino, Toshio Sone, and Ken'iti Kido, "Performance evaluation in speech recognition system using transition probability between linguistic units," in *Proc. Int. Conf. Spoken Language Processing*, (Kobe, Japan), pp. 1231–1234, Nov. 1990.

[150] Bhadriraju Krishnamurthi, *The Dravidian Languages.* Cambridge, U.K: Cambridge university press, 2003.

[151] Norman Abramson, *Information Theory and Coding.* New York: McGraw Hill, 1963.

[152] E. J. Yannakoudakis and G. Angelidakis, "An insight into the entropy and redundancy of the English dictionary," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 960–970, Nov. 1988.

[153] Steve Young, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland, *The HTK Book Version 2.2.* New Jersy: Entropic, 1999.

[154] C. Chandra Sekhar, W. F. Lee, K. Takeda, and F. Itakura, "Acoustic modeling of subword units using support vector machines," in *Workshop on Spoken Language Processing*, (TIFR, Mumbai, India), pp. 79–86, Jan. 2003.

[155] S. V. Gangashetty, C. Chandra Sekhar, and B. Yegnanarayana, "Dimension reduction using autoassociative neural network models for recognition of consonant-vowel units of speech," in *Proc. Fifth Int. Conf. Advances in Pattern Recognition*, (ISI Calcutta, India), pp. 156–159, Dec. 2003.

[156] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, Jan. 1995.

[157] B. Yegnanarayana and S. P. Kishore, "AANN: An alternative to GMM for pattern recognition," *Neural Networks*, vol. 15, pp. 459–469, Apr. 2002.

[158] M. S. E. Alexander Strehl, *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining.* PhD thesis, The University of Texas at Austin, May 2002.

[159] B. Yegnanarayana, K. Sharat Reddy, and S. P. Kishore, "Source and system features for speaker recognition using AANN models," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (Saltlake City, USA), pp. 409–412, May 2001.

[160] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, Mar. 1998.

List of Publications

1. A. Nayeemulla Khan and B. Yegnanarayana, "Vowel onset point based variable frame rate analysis for speech recognition," in *Proc. IEEE Int. Conf. on Intelligent Sensing and Information Processing*, (Chennai, India), Jan. 2005.

2. A. Nayeemulla Khan, "Syllable as a unit for speech recognition: An information theoretic perspective," in *Proc. Int. Conf. Natural Language Processing*, (Hyderabad, India), pp. 219–224, Dec. 2004.

3. A. Nayeemulla Khan and B. Yegnanarayana, "Latent semantic analysis for speaker recognition," in *Proc. Int. Conf. Spoken Language Processing*, vol. 4, (Jeju, Korea), pp. 2589–2592, Oct. 2004.

4. A. Nayeemulla Khan and B. Yegnanarayana, "Speech enhanced multi-span language model," in *Proc. Int. Conf. Spoken Language Processing*, vol. 3, (Jeju, Korea), pp. 2249–2252, Oct. 2004.

5. A. Nayeemulla Khan, Suryakanth V. Gangashetty, and B. Yegnanarayana, "Neural network preprocessor for recognition of syllables," in *Int. Conf. on Intelligent Sensing and Information Processing*, (Chennai, India), pp. 171–175, Jan. 2004.

6. A. Nayeemulla Khan, Suryakanth V. Gangashetty, and B. Yegnanarayana, "Syllabic properties of three Indian languages: Implications for speech recognition and language identification," in *Proc. Int. Conf. Natural Language Processing* (Communicated, ed.), (Mysore, India), pp. 125–134, Dec. 2003.

7 A. Nayeemulla Khan, "Recognition of syllable like units in Indian languages," in *Proc. Int. Conf. Natural Language Processing*, (Mysore India), pp. 135–141, Dec. 2003.

8 Suryakanth V. Gangashetty, K. Sreenivasa Rao, A. Nayeemulla Khan, C. Chandra Sekhar, and B. Yegnanarayana, "Combining evidence from multiple modular networks for recognition of consonant-vowel units of speech," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 1, (Portland, Oregon, USA), pp. 686–691, July 2003.

9 K. Srinivasa Rao, Suryakanth V. Gangashetty, and A. Nayeemulla Khan, "Distribution capturing ability of autoassociative neural network models for recognition of Consonant-Vowel utterances," in *Proc. Seventh Int. Conf. Cognitive and Neural Systems*, vol. 1, (Boston USA), p. 28, May 2003.

10 A. Nayeemulla Khan, Suryakanth V. Gangashetty, and S. Rajendran, "Speech database for Indian languages- A preliminary study," in *Proc. Int. Conf. Natural Language Processing*, (NCST, Mumbai, India), pp. 295–301, Dec. 2002.

11 Gaurav Aggarwal, Anvita Bajpai, A. Nayeemulla Khan, and B. Yegnanarayana , "Exploring features for audio indexing," in *IRISS '02*, (Indian Inst. of Sci., Bangalore, India), 2002.

12 Suryakanth V. Gangashetty, A. Nayeemulla Khan, S. R. Mahadeva Prasanna, and B. Yegnanarayana, "Neural network models for preprocessing and discriminating utterances of consonant-vowel units," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 1, (Honolulu, Hawaii, USA), pp. 613–618, May 2002.

13 A. Nayeemulla Khan and B. Yegnanarayana, "Development of a speech recognition system for Tamil for restricted small tasks," in *National Conf. Commn.*, (IIT Kanpur), Jan. 2001.

CURRICULUM VITAE

1. **Name:** A. Nayeemulla Khan
2. **Date of birth:** 03-Aug-1969
3. **Permanent address:** New 5A, old 33A, Mahalakshmi Colony Thiruvalluvar Salai, Thiruvanmiyur, Chennai 600 041.
4. **Educational qualifications:**

   - 1990, Bachelor of Engineering (B.E., Civil Engineering, Madras University, Chennai, India).
   - 1992, Master of Engineering (M.E., Structural Engineering, Annamalai University, Tamilnadu).
   - 1995, Diploma in Computer Applications Scientific and Engineering, (DCA, National Center for Computing Techniques, Chennai).
   - 2002, Master of Science (M.S, Information Systems and Applications, Bharathidasan University, Trichy)
   - 2006, Doctor of Philosophy (Ph.D, Computer Science and Engineering Department, IIT Madras, India)

DOCTORAL COMMITTEE

- **Chairperson:** Prof. Timothy A. Gonsalves
- **Guide:** Prof. B. Yegnanarayana
- **Members:**

1. Prof. K. Radhakrishna Rao (Dept. of Electrical Engg.)
2. Prof. Sreesh C. Chaudhary (Dept. of Humanities and Social Sciences)
3. Dr. Deepak Khemani (Dept of Computer Science and Engg.)
4. Dr. Hema A. Murthy (Dept of Computer Science and Engg.)