IDIAP RESEARCH REPORT

USING AUXILIARY SOURCES OF KNOWLEDGE FOR AUTOMATIC SPEECH RECOGNITION

Mathew Magimai Doss

IDIAP–RR 05-90

JULY 2005

PUBLISHED IN
EPFL Thesis No. 3263

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 05-90

USING AUXILIARY SOURCES OF KNOWLEDGE FOR AUTOMATIC SPEECH RECOGNITION

Mathew Magimai Doss

JULY 2005

PUBLISHED IN
EPFL Thesis No. 3263

Abstract

Standard hidden Markov model (HMM) based automatic speech recognition (ASR) systems usually use cepstral features as acoustic observation and phonemes as subword units. Speech signal exhibits wide range of variability such as, due to environmental variation, speaker variation. This leads to different kinds of mismatch, such as, mismatch between acoustic features and acoustic models or mismatch between acoustic features and pronunciation models (given the acoustic models). The main focus of this work is on integrating auxiliary knowledge sources into standard ASR systems so as to make the acoustic models more robust to the variabilities in the speech signal. We refer to the sources of knowledge that are able to provide additional information about the sources of variability as auxiliary sources of knowledge. The auxiliary knowledge sources that have been primarily investigated in the present work are auxiliary features and auxiliary subword units.

Auxiliary features are secondary source of information that are outside of the standard cepstral features. They can be estimation from the speech signal (e.g., pitch frequency, short-term energy and rate-of-speech), or additional measurements (e.g., articulator positions or visual information). They are correlated to the standard acoustic features, and thus can aid in estimating better acoustic models, which would be more robust to variabilities present in the speech signal. The auxiliary features that have been investigated are pitch frequency, short-term energy and rate-of-speech. These features can be modelled in standard ASR either by concatenating them to the standard acoustic feature vectors or by using them to condition the emission distribution (as done in gender-based acoustic modelling). We have studied these two approaches within the framework of hybrid HMM/artificial neural networks based ASR, dynamic Bayesian network based ASR and TANDEM system on different ASR tasks. Our studies show that by modelling auxiliary features along with standard acoustic features the performance of the ASR system can be improved in both clean and noisy conditions.

We have also proposed an approach to evaluate the adequacy of the baseform pronunciation model of words. This approach allows us to compare between different acoustic models as well as to extract pronunciation variants. Through the proposed approach to evaluate baseform pronunciation model, we show that the matching and discriminative properties of single baseform pronunciation can be improved by integrating auxiliary knowledge sources in standard ASR.

Standard ASR systems use usually phonemes as the subword units in a Markov chain to model words. In the present thesis, we also study a system where word models are described by two parallel chains of subword units: one for phonemes and the other are for graphemes (phoneme-grapheme based ASR). Models for both types of subword units are jointly learned using maximum likelihood training. During recognition, decoding is performed using either or both of the subword unit chains. In doing so, we thus have used graphemes as auxiliary subword units. The main advantage of using graphemes is that the word models can be defined easily using the orthographic transcription, thus being relatively noise free as compared to word models based upon phoneme units. At the same time, there are drawbacks to using graphemes as subword units, since there is a weak correspondence between the grapheme and the phoneme in languages such as English. Experimental studies conducted for American English on different ASR tasks have shown that the proposed phoneme-grapheme based ASR system can perform better than the standard ASR system that uses only phonemes as its subword units. Furthermore, while modelling context-dependent graphemes (similar to context-dependent phonemes), we observed that context-dependent graphemes behave like phonemes. ASR studies conducted on different tasks showed that by modelling context-dependent graphemes only (without any phonetic information) performance competitive to the state-of-the-art context-dependent phoneme-based ASR system can be obtained.

Version abrégée

Les systèmes de reconnaissance de la parole (ASR) utilisant des chaînes de Markov cachées (HMM) utilisent généralement des données cepstrales comme observations, et des phonèmes comme modèles elémentaires. Pour le même contenu lexical, le signal de parole est très variable. La variabilité peut être causée par l'environnement ou la personne. Ceci conduit à différent types de disparités : entre observations et modèles acoustiques, ou bien entre observations et modèles de prononciation (connaissant les modèles acoustiques). L'objet principal de cette thèse est d'incorporer des sources d'informations "auxiliaires" dans les systèmes standards d'ASR, pour les rendre plus robustes à la variabilité du signal de parole. Ces sources d'informations auxiliaires apportent une connaissance additionnelle sur les sources de variabilité, comme par exemple la vitesse d'élocution. Les principales sources d'informations auxiliaires considérées dans cette thèse sont de deux types : les observations auxiliaires et les modèles auxiliaires des phonèmes.

Les observations auxiliaires apportent une connaissance complémentaire aux observations cepstrales. Elles peuvent être estimées directement à partir du signal de parole (timbre, énergie, vitesse d'élocution), ou bien à partir de mesures complémentaires (position de la machoire, information visuelle). Etant corrélées avec les observations acoustiques standards, elles peuvent permettre de construire de meilleurs modèles acoustiques, en les rendant moins sensibles à la variabilité du signal de parole. Dans cette thèse, nous avons étudié le timbre, l'énergie et la vitesse d'élocution. Ces observations auxiliaires sont intégrées à un système standard d'ASR, soit en concaténant observations acoustiques et auxiliaires, soit en utilisant les observations auxiliaires pour conditionner les probabilités d'émission des observations acoustiques. Nous avons étudié ces deux approches appliquées à trois type de systèmes : système hybride HMM/ANN (réseau neuronal), système HMM/DBN (résau dynamique bayésien), et système TANDEM. Plusieurs tâches d'ASR sont considérées. Les résultats montrent que les observations auxiliaires permettent d'améliorer la performance d'ASR, à la fois dans les environnement bruités et non-bruités.

De plus, nous proposons une approche d'évaluation de la prononciation de base de chaque mot, vis-à-vis des données observées. Cette approche permet à la fois d'extraire automatiquement de nouveaux modèles de prononciation, de les comparer entre elles et d'évaluer la stabilité de la prononciation de base. L'information auxiliaire apportée par les nouvelles prononciations permet d'améliorer la performance d'ASR.

Enfin, cette thèse étudie la modélisation acoustique en terme de graphèmes, comme complément à la modélisation standard, faite en terme de phonèmes. Pour modéliser un mot, deux chaînes parallèles de sous-unités — phonèmes et graphèmes — sont utilisées. L'apprentissage se fait de façon conjointe, pour maximiser la vraisemblance des données observées. Pendant la reconnaissance, le décodage est fait en utilisant soit l'un des deux types de modèles, soit les deux ensembles. Dans tous les cas, la modélisation par graphèmes est utilisée commme information auxiliaire. L'avantage principal des graphèmes est que chaque mot peut être modélisé facilement en utlisant la transcription orthographique. Celle-ci peut être considérée comme moins bruitée par rapport à la transcription en termes de phonèmes. Cependant, les graphèmes ont un désavantage, car la correspondance entre entre graphèmes et phonèmes est faible dans certains langages comme l'Anglais. Les résultats d'ASR pour l'Anglais américain sur différentes tâches montrent que l'adjonction de l'information auxiliaire des graphèmes peut améliorer la performance de la reconnaissance. De plus, lors de l'étude des modèles à base de graphèmes dépendant du contexte, nous observons qu'ils se comportent de façon similaire aux phonèmes. Les performances d'ASR des systèmes utilisant uniquement des graphèmes dépendant du contexte, sans information phonétique, sont similaires aux performances des systèmes habituels, qui utilisent des phonèmes dépendant du contexte.

Contents

List of Figures

List of Tables

Acknowledgments

I express my sincere gratitude to my thesis director, Hervé Bourlard for the guidance, collaboration and support that culminated in this thesis. Hervé introduced me to core speech recognition research. During the course of my thesis work, I learned a lot from his analytical and systematic approach to scientific research. I would also like to thank him for sharing with me his wisdom, knowledge, "funny" stories, jokes and, his vision and enthusiasm for research.

I would like to thank Hynek Hermansky for the collaboration, guidance and, for sharing his knowledge and understanding of speech science. Hynek's statement during speech group meetings "how come you all keep showing always improved recognition performance and no negative results." still rings a bell in the mind.

I would like to acknowledge the people from India who were important in getting me started in graduate school. Prof. B. Yegnanarayana my M.S supervisor at IIT Madras, Dr. K. Samudravijaya from TIFR Bombay, and Suresh Kumar my undergraduate advisor and T. G. Srinivasulu my undergraduate mathematics professor at Adhiyamaan College of Engineering Hosur.

I want to thank the past and present members of IDIAP. I am lucky to know, work, collaborate, socialize and cultivate friendship with such a great group of people. I have learned a lot from many group meetings, discussions and collaborations with colleagues working in different areas of research at IDIAP. In particular this work has also benefited directly from collaborations with Aissa Ait-Hassou, Samy Bengio, John Dines, Jaume Escofet Carmona, Petr Fousek, Frantisek Grezl, Shajith Ikbal, Guillaume Lathoud, Bertrand Mesot, Hemant Misra, Sunil Sivadas and Todd Stephenson. I would like to thank the members of Speech Group at ICSI Berkeley, particularly Barry Chen, Nelson Morgan and Qifeng Zhu for sharing their experiences with large vocabulary automatic speech recognition systems and, for their collaboration and help. I would also like to thank my thesis jury who provided lots of valuable inputs for improving this thesis. Guillaume Lathoud provided his French translation services.

Doing PhD would have been much difficult without friends who always brought refreshing moments through their moral support, their enthusiasm to know about my research and its practical implication, and sharing their leisure hours having stimulating discussions, meals, outings etc. To mention a few, I would like to thank Anil, Anita, Anja, Balan, Bhavna, Benny, Chirdeep, Christelle, Geetha, Gnana, Jahnavi, Jasmine, Jordi (and his family), John Navin, Justin Sagayaraj, Jyotsna, Kishore, Muralidharan, Murthy, Murugan, Partha, Rams, Roseena, Soumaya, Vidhya, and Xavier.

I would like to thank my family and my wife Francina's family for their encouragement and support.

Finally I would like to thank Francina, with out her patience, support and aid this thesis would not have been completed.

This work was carried out in the framework of the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). The NCCR are managed by the Swiss National Science Foundation on behalf of the federal authorities. This work was also possible through financial support both from the Swiss National Science Foundation under the grant PROMO (21-57245.99) and from IDIAP.

Chapter 1

Introduction

1.1 Objective of the Thesis

The ultimate goal of automatic speech recognition (ASR) is to recognize the spoken message irrespective of who is speaking, when, where, and how. In the last few decades, speech recognition systems with limited capabilities have been available commercially. The performance of these systems vary as a function of the transducer (microphone to telephone), vocabulary (small to medium size), speaker (speaker dependent to speaker independent) and operating environment (clean to noisy conditions). In other words, the performance of these systems drop as the constraints are relaxed, such as, going from single speaker to multiple speakers, clean conditions to noisy conditions, medium size vocabulary to large vocabulary continuous speech, microphone speech to telephone speech. The main reason behind this is the presence of additional variabilities in the speech signal. The variabilities in speech arise from several factors, e.g., differences in the production mechanism of speakers, differences in the environmental, and channel conditions etc. Present ASR systems have limited ability to handle these variabilities. Thus, current ASR researches focus on developing ASR systems that are insensitive to the variabilities in the speech signal and attain human like recognition performance.

The main objective of this thesis is to integrate auxiliary sources of knowledge in state-of-the-art ASR system for improved performance. We refer to the sources of knowledge that are able to provide additional information about the sources of variability as auxiliary sources of knowledge. In other words, auxiliary knowledge sources bring additional information complementary to the standard acoustic features or models which helps in reducing variability and improving ASR performance.

1.2 Automatic Speech Recognition

In the early days of ASR research, knowledge-driven approaches were more prominent (Reddy, 1967; Klatt, 1977; Lesser *et al.*, 1975; Zue, 1985; Holmes and Huckvale, 1994). For a review of knowledge-based systems in speech recognition refer to (Klatt, 1977; O'Brien, 1993). In these systems, the acoustic signal was first segmented and labeled into phoneme-like units, followed by rule-based lexical and syntactic analysis. The main emphasis was on applying artificial intelligence techniques to use higher level knowledge lexicon, syntax, semantics, and pragmatics. These systems were computationally expensive, task-dependent, and yielded quite poor recognition performance. The poor recognition performance was partially related to the difficulty in reliably extracting phonetic information from the speech signal (Zue, 1985).

Later, pattern matching approaches came into prominence. Two popular pattern matching approaches, namely, dynamic time warping based (deterministic pattern matching) and hidden Markov models (HMMs) based (stochastic pattern matching) emerged (Sakoe and Chiba, 1978; Bridle *et al.*, 1983; Baker, 1975; Jelinek, 1976; Bahl *et al.*, 1983; Levinson, 1985). Today, state-of-the-art ASR systems are based on HMM. As opposed to artificial intelligence based, HMMs are based on "ignorance modelling", i.e., the parameters of a model are automatically trained using a large amount of training data, but very little speech knowledge (Gevins and Morgan, 1984; Makhoul and Schwartz, 1985). The other advantages are that HMM-based ASR is flexible (e.g. choice of acoustic analysis), allows application of speech knowledge by specifying topological constraints, and has tractable mathematical structure (Jelinek, 1976; Bahl *et al.*, 1983; Levinson, 1985; Rabiner, 1989).

In HMM-based ASR systems, the states in the HMM model the short-term spectral characteristics and, the sequence of the states model the temporal relationship (Markov chain). The spectral characteristics are represented by acoustic feature vector usually consisting of cepstral coefficients and their derivatives. The states usually represent subword units and, each state is associated with a probability distribution/density function of the acoustic feature vectors. The word models are constructed by concatenating the subword models according to prior knowledge (e.g., phonetic transcription from the lexical dictionary). The HMM-based approach is a pattern matching approach. It involves two steps, namely, training and testing. During training the parameters of the models (e.g., parameters of probability distribution) are learned. During testing, unlike the knowledge-based approaches acoustic, lexical, and syntactical analysis is performed jointly and, the output of the recognition process is a sequence of words. The probabilistic framework in HMM-based ASR allows the system to generalize to unseen data to a certain extent. The HMM-based ASR has a principled way to integrate all knowledge sources, as long as they can be formulated in statistical terms. For instance, in HMM-based ASR the words are composed of sequence of states (subword units) and, words are sequence of states for higher syntactic level and, the syntactic level can be part of higher semantic level. The outcome of pattern matching in HMM-based ASR is a best match at each level, not involving any local decision. HMM-based ASR systems have thus been more successful than the knowledge-based ASR systems.

1.3 Why Auxiliary Sources of Knowledge?

Although HMM has been the basis for most successful ASR systems, the performance is still far from ideal human-like recognition for complex tasks, such as, large vocabulary or continuous spontaneous speech or recognition in adverse conditions. The main reason for this is the undesirable variabilities present in the speech signal. The different sources of variability are:

1. Speaker: While speaking due to different reasons human introduce variability. For instance, to communicate well they may change their emotions or in noisy background conditions they may speak louder. The speaker variability can be further categorized as with-in speaker or across-speakers, where, the earlier source of variability is referred to as intra-speaker variability and, the later as inter-speaker variability.

    - Intra-Speaker speaker variability results from the changes in the speaker's physical and emotional state, speaking rate and voice quality.
    - Inter-Speaker variability can result from differences in the dialect, accent, physiology (length and shape of vocal tract), and voice source characteristics (e.g., pitch frequency).

2. Transducer and Channel: The speech signal can be collected through different transducers and can be transmitted through different channels, e.g., microphone speech and telephone speech. The characteristics (e.g, transfer function, gain, bandwidth) of different transducers and transmission channels greatly differ. For instance, the telephone speech is band limited between 300 Hz and 3300 Hz, whereas the microphone speech has a higher bandwidth. So, during speech data collection the characteristic of the transducer and the transmission channel influence the speech signal bringing in more variability.

3. Environment: The input to the transducer sensing the speech signal is not only the acoustic pressure waves emitted by the speech production system of intended speaker, but also the noise signal from the surrounding environment. The noise signal consists of signal from other sound sources in the surrounding environment such as, human speaking in the background or traffic noise and time-delayed versions of the intended speaker's speech signal due to reverbation. The noise signal interferes with the speech signal resulting in additional variability. Furthermore, the background noise can influence the speech generation itself such as, the intended speaker raises vocal intensity changing the characteristic of the speech signal. This is called Lombard effect (Junqua, 1993).

4. Phonetic: Standard ASR system usually use phonemes as subword units. The acoustic realization of the phonemes are highly dependent upon the adjacent phonemes. For instance, the acoustic realization of phoneme /a/ in word cat /k/ /a/ /t/ and word bat /b/ /a/ /t/ is different. This is mainly due to coarticulation. Coarticulation may be generally defined as "the overlapping of adjacent articulations". The result of coarticulation and other phonological processes, such as, assimilation, reduction, insertion, and deletion lead to pronunciation variation. Pronunciation variation also results due to speaker variability.

The variabilities present in the speech signal influences the ASR system at various levels. For instance, the transmission channel variabilities has an effect on the acoustic feature vectors, where as, the pronunciation variation has effect upon both the acoustic feature vector and the lexical model. One way to reduce the effect of these variabilities is integrating auxiliary sources of knowledge in the state-of-the-art HMM-based ASR systems.

Auxiliary sources of knowledge are able to provide additional information to reduce variability. For example, an artificial neural network (ANN) trained to classify phoneme with pitch frequency as input feature may yield 30-40% phoneme recognition accuracy and, this may not be sufficient for speech recognition. However, pitch frequency can convey different information, such as, gender of a person, emotional state of a person, and the intonation pattern. Thus, by integrating auxiliary sources of knowledge into HMM-based ASR system we can expect to provide the system with additional knowledge or capability to handle variation present in the speech signal.

There are different issues when integrating auxiliary sources of knowledge, including

1. What kind of auxiliary sources of knowledge should be used?

2. How they should be integrated in the state-of-the-art HMM-based ASR system?

Previous research studies on acoustic-phonetic, speech perception, knowledge-based ASR, and other complimentary areas of speech technology (e.g., speech synthesis, speaker recognition) can help us in identifying the auxiliary sources of knowledge. The issue of how to integrate the auxiliary source of knowledge depends upon the availability of a tractable mathematical formulation with out drastically increasing the complexity of the system. In other words, the integration of auxiliary sources of knowledge in HMM-based ASR system can be seen as a mechanism to integrate the speech knowledge in systematic and controlled manner.

1.4 Contribution of the Thesis

The central theme of this thesis is the integration/use of auxiliary sources of knowledge in state-of-the-art HMM-based ASR systems. We have investigated two different types of auxiliary source of knowledge:

- **Auxiliary Feature**: Auxiliary features are secondary sources of information that are outside of the standard acoustic features. They can be used to:
  - reduce the variability of the representation

- improve the matching properties of the model.

The auxiliary features that have been investigated in this thesis are pitch frequency, short-term energy and rate-of-speech. All these auxiliary features are directly estimated from the speech signal. We have studied two main ways to integrate them into HMM-based ASR system. In the first approach, the auxiliary feature is treated like standard cepstral feature by concatenating it with the standard cepstral feature. In the second approach, the auxiliary feature is used to condition the acoustic model (conditioning the emission distribution). Both of these approaches have been studied for different speech recognition tasks, namely, isolated word recognition, connected word recognition and continuous speech recognition with different variations of HMM-based systems.

- **Auxiliary Subword Unit**: Standard HMM-based ASR systems usually use phonemes as subword units. We extend the notion of auxiliary sources of knowledge to model more than one subword units, where, subword units other than the phonemes act as auxiliary subword units. We have used graphemes as the auxiliary subword units for English language. We have proposed a phoneme-grapheme based ASR system that jointly models the phoneme and grapheme subword units.

Model Evaluation: In this thesis, we have proposed an approach to evaluate the adequacy of the pronunciation model in terms of matching and discriminating properties of single baseform pronunciation. Through the proposed approach to evaluate baseform pronunciation, we show that integrating auxiliary sources of knowledge improves the “stability” of the baseform pronunciation.

1.5 Organization of the Thesis

Chapter 2 gives an overview of the state-of-the-art speech recognition system. We describe the different steps involved in automatic speech recognition, such as, feature extraction, acoustic modelling, pronunciation modelling, language modelling, and decoding.

In Chapter 3, we briefly describe the HMM-based ASR and, present the different state-of-the-art HMM-based ASR systems that have been used in our studies. We describe the experimental setup of different speech recognition tasks that have been addressed in this thesis along with the measures that have been used to evaluate these systems.

Chapter 4 describes the integration of auxiliary knowledge sources in the state-of-the-art ASR system. We introduce the notion of auxiliary feature and relate it to previous research works in this direction. In this chapter, different approaches to model (concatenation and conditioning) the auxiliary features along with standard features in HMM-based ASR system have been proposed. These approaches are evaluated for different ASR tasks using context-independent phonemes as subword units. In Chapter 5, we present the ASR studies evaluating the proposed approach using context-dependent phonemes as subword units.

Chapter 6 presents the approach to evaluate pronunciation models of words and pronunciation variant extraction. We compare the proposed pronunciation model evaluation approach with the acoustic models trained only with standard cepstral features with acoustic models trained with both standard and auxiliary features.

Chapter 7 presents the phoneme-grapheme based ASR system, where the phoneme subword units and grapheme subword units are jointly modelled. The grapheme is used as auxiliary subword unit. We describe the modelling approach in phoneme-grapheme based ASR and present studies on different ASR tasks. While studying phoneme-grapheme based ASR system, we found that modelling grapheme contextual information can yield performance similar to state-of-the-art ASR systems using phoneme as subword units. This motivated us to further look into modelling graphemic context for ASR. We also present these studies on different ASR tasks.

Chapter 8 summarizes the present work and presents the conclusion of this thesis work along with future directions.

Chapter 2

Speech Recognition Fundamentals

2.1 Introduction

Given an acoustic sequence $X = \{x_1, \cdots x_n, \cdots x_N\}$, the goal of ASR is to extract the most probable lexical representation W. The acoustic sequence X is the parametric representation of the speech signal[1]. From the information theoretic point of view, ASR may be considered as a bit reduction problem. At the input of the speech recognition, system we have an information rate of 128 Kb/s (one second telephone speech signal sampled at 8000 Hz represented by 2 bytes per sample) and, at the output of the speech recognition system the linguistic message which has an information rate of approximately 50-60 b/s (12-14 phonemes per second in normal human speech and Huffman coding). This problem is then formulated statistically as finding the word sequence W which is most likely to have produced X:

$$\hat{W} = \arg\max_{W} P(W|X, \Theta) \tag{2.1}$$

where Θ are the parameters of the system. Direct estimation of the probability $P(W|X, \Theta)$ is practically infeasible (as discussed later in this section), hence rewriting the above equation by applying Bayes rule yields:

$$\begin{aligned} \hat{W} &\simeq \arg\max_{W} \frac{p(X|W, \Theta_a)P(W|\Theta_l)}{p(X|\Theta)} \\ &\simeq \arg\max_{W} p(X|W, \Theta_a)P(W|\Theta_l) \end{aligned} \tag{2.2}$$

where, $\Theta = \{\Theta_a, \Theta_l\}$. Since we are interested in maximization over word sequences W, the denominator $p(X|\Theta)$ is irrelevant when making decision, thus is omitted. In the literature, $p(X|W, \Theta_a)$ is referred to as the acoustic model and Θ_a are the parameters of the acoustic model, while $P(W|\Theta_l)$ is referred to as the language model and Θ_l are the parameters of the language model (Jelinek, 1976; Bahl *et al.*, 1983; Jelinek, 1997). The speech recognition system is parameterized by these two models, $\Theta = \{\Theta_a, \Theta_l\}$.

On the application side of speech recognition, there are a variety of common tasks, such as isolated word recognition, connected word recognition, and continuous speech recognition. In the case of isolated word recognition, the complexity of the task is much lower as the objective is the recognition of a single word as opposed to a sequence of words. This task can be directly realized via (2.1). The prior distribution $p(X|\Theta)$ can be simply estimated over a finite set of words as $\sum_{i=1}^{L} P(X|W_i, \Theta_a)P(W_i|\Theta_l)$, where L is the finite number of isolated words. However, for connected word recognition tasks and continuous speech recognition tasks such an approach cannot be

[1] Later in this chapter, we describe the process of parametric representation of the speech signal

adopted as many samples of all possible word sequences W will be required. This is practically infeasible, hence taking an alternate approach of combined classification and segmentation for these tasks. An example of a connected word recognition task is the recognition of a sequence of numbers. In such a case, the definition of $P(W|\Theta_l)$ is quite easy as any word (number) can follow any word. The continuous speech recognition task is more related to the spoken language where, there is a greater influence of the grammar of the language, i.e., certain sequence of words are more likely than others. Thus, for a continuous speech recognition task the estimation of $P(W|\Theta_l)$ is not straight-forward (discussed later in Section 2.5).

In this chapter, we briefly summarize the different components of an ASR system. Figure 2.1 describes the different components of an ASR system.

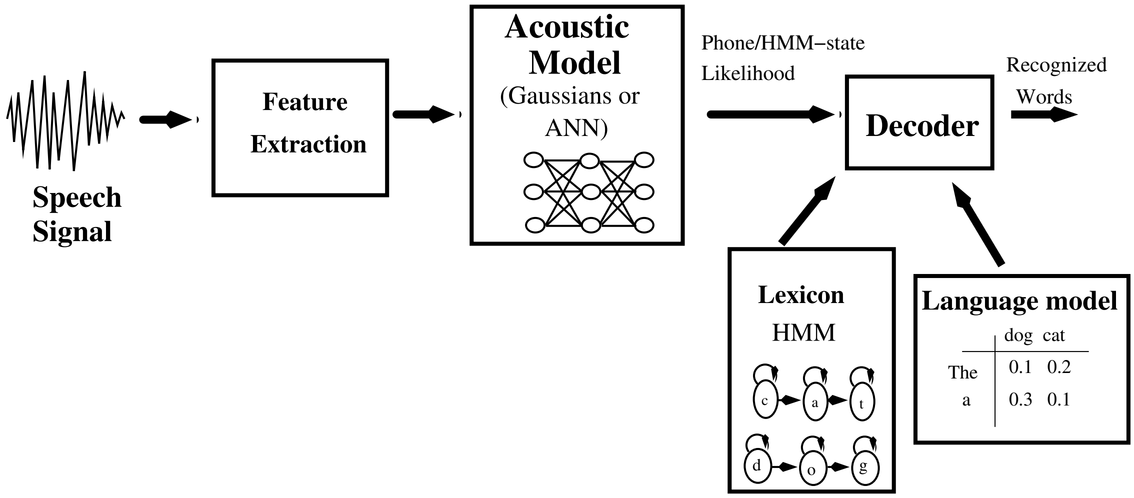


Figure 2.1. Block diagram of ASR system.

The speech signal carries lots of different information much of which is highly redundant. The first major step is to discard the "irrelevant" part of the speech signal and, extract only the information which represents the "essence" of the underlying message. This process is called feature extraction. We describe the process of speech signal analysis and feature extraction in Section 2.2. The result of feature extraction process is a sequence of acoustic feature vectors $X = \{x_1, \cdots x_n, \cdots x_N\}$. The acoustic modelling stage models this sequence of acoustic feature sequence X to build reference models. Section 2.3 describes the acoustic modelling process (i.e., estimation of parameters Θ_a) in the state-of-the-art ASR. The reference models can be of words or of units shorter than words. Since there are a large number of words in a single language, it is difficult to collect enough acoustic realizations of each and every word. Smaller phonological recognition units (also called subword units) such as, phonemes or syllables are used to overcome this problem. This process of mapping from words to subword units is described in Section 2.4. The lexicon of an ASR system contains the word and its transcription in terms of subword units. Section 2.5 briefly summarizes the standard language modelling techniques. The final stage of a speech recognition system is decoding, where the acoustic model and language model are combined in order to find the most likely sequence of words. We briefly describe the process of decoding in Section 2.6.

2.2 Speech Signal Analysis and Feature Extraction

Speech is produced by the excitation of the time varying vocal tract system by a time varying signal. The excitation is generated by air flow from the lungs carried by the trachea through the vocal cords. As acoustic waves pass through the vocal tract, the frequency content (spectrum) is modulated by the resonances of the vocal tract. The output of the vocal tract are pressure waves which are then detected by the human ear and, converted to an electrical signal and, further processed by the human brain. The first stage of the speech signal analysis is speech acquisition. In speech acquisition, the acoustic waves emitted by the vocal tract system are captured by a microphone and subsequently converted from analog signal into digital signal by sampling (higher than Nyquist rate) and quantization (16 bits representation). When the speech is collected over a telephone channel, the speech signal is band limited between 300-3300 Hz, hence, the speech signal is sampled at 8000 Hz. In case of microphone speech, the sampling frequency can vary from 11.025 kHz up to 20 kHz.

2.2.1 Analysis of Speech Signal

In ASR systems, the digital speech signal is typically modelled as the convolution of excitation signal $e(t)$ and vocal tract response $h(t)$:

$$s(t) = e(t) * h(t) \tag{2.3}$$

As mentioned earlier in this chapter, ASR is a data reduction process. As a first stage of data reduction the ASR system usually converts speech into a spectral representation (by Fourier transform), consisting of acoustic features. The amplitude spectrum[2] of the speech signal is then represented as:

$$|S(\omega)| = |E(\omega)| \cdot |H(\omega)| \tag{2.4}$$

The spectral representation of the speech signal is much more informative for speech sound discrimination than the time domain signal. This is also supported by the findings in speech perception that the human ear performs a non-linear frequency analysis (O'Shaughnessy, 1987; Rabiner and Juang, 1993). In $|S(\omega)|$, the fine structure results from $|E(\omega)|$ and, the envelope is defined by $|H(\omega)|$. The vocal tract excitation $|E(\omega)|$ features are voicing, amplitude, and pitch frequency[3] which are more influenced by higher level linguistic phenomena (e.g., syntax and semantic structure) than by the individual sounds being produced. The spectral envelope $|H(w)|$ embodies the vocal tract resonances referred to as formants, of which the location and bandwidth are more representative of the sound (phoneme) being produced. Thus, most ASR systems parameterize the spectral envelope in the form of a 8 to 14 dimensional feature vector. The spectral envelope can be characterized by linear prediction (LP) parameters Makhoul (1975) and their transformations or by cepstrum. The cepstrum deconvolves the vocal tract response $h(n)$ from the excitation $e(n)$:

$$\log(|S(\omega)|) = \log(|E(\omega)|) + \log(|H(\omega)|), \tag{2.5}$$

$$\mathrm{IFT}(\log(|S(\omega)|) = \mathrm{IFT}((\log|E(\omega)|)) + \mathrm{IFT}(\log(|H(\omega)|)),$$

$$c(t) = \hat{s}(t) = \hat{e}(t) + \hat{h}(t), \tag{2.6}$$

where, IFT means inverse Fourier transform. The cepstrum separates approximately into a slow component $\hat{h}(t)$ corresponding to the envelope and, a fast component $\hat{e}(t)$ for vocal tract excitation. It is this low-time component, the initial set of 8 to 14 values of $c(t)$, that is used as a feature for the ASR system. These features are called cepstral features. In the following section, we describe two different cepstral features that are mainly used in state-of-the-art ASR systems.

The spectrum of the speech signal is highly imbalanced because of spectral roll off, i.e., there is more energy in the low frequency bands than the high frequency bands. Consequently, the information in lower frequency bands are better represented than higher frequency bands. In order

[2] It is commonly assumed that any residual information in the phase is redundant and can be ignored (O'Shaughnessy, 2003).

[3] Pitch is a perceptual quantity, but its acoustic correlate (rate of vibration of vocal cords), referred to as fundamental frequency, can be estimated from the speech signal. In this thesis, we refer to the fundamental frequency as pitch frequency.

to handle this spectral imbalance, the speech signal is passed through a pre-emphasis filter which flattens the spectrum i.e., tilts the spectrum upwards with increasing frequency. The filter simply replaces each sample $s(t)$ with its differenced version $s(t) - as(t-1)$, where the a is typically 0.95.

Most of the spectral analysis algorithms, such as Fourier transform, assume that the signal is stationary. In reality speech is a nonstationary signal, thus, speech analysis is carried out by applying window of shorter duration (also called frame) in which the speech signal can be assumed quasi-stationary. The feature is then computed in each frame, resulting in a sequence of feature vectors X. The window is applied by multiplying the speech signal with a window function $w[n]$ of M nonzero samples (M is the window length, also called frame size). Since the window is multiplied with the speech signal, the frequency response of the window function is convolved with the speech spectrum. In order to minimize the effect of the window function on the original spectrum of the speech signal, the selection of window function is important. The selection of window is a trade-off between frequency selectivity (determined by the width of the main lobe in frequency domain) and energy leakage (determined by the discontinuities at the edge of the window). For a given finite window length M, the Hamming window has low frequency selectivity compared to rectangular window; but the energy leakage is lower for Hamming window in comparison to rectangular window. Hence, Hamming window is commonly used as the windowing function. Since the Hamming window tapers at their edges, they heavily attenuate the samples at the edges. If there was no overlap between the windows across time then, the analysis would leave over look certain number of speech samples. The length of the overlap is determined by the update interval P (also called frame shift), which reflects how often the ASR system evaluates the portion of the speech signal. The selection of the frame size M and frame shift P is very crucial. If the frame size is large then it would lead to smearing of the dynamic effects of vocal tract transitions, and if the frame size is too small then the updates are very frequent, thus risking incorporating pitch frequency information in feature sequences. Standard ASR systems use frame size of 25-30 ms with a frame shift of 10-20 ms. This ensures that there are at least two pitch periods when analyzing voiced speech and, there are 50 to 100 frames/s reflecting the velocities of the vocal tract movements (20 Hz - 50 Hz) relevant to phone identification (O'Shaughnessy, 2003).

2.2.2 Standard Acoustic Features

At each time frame we extract acoustic feature vectors, typically cepstral features as described earlier. The most common acoustic features used in state-of-the-art ASR systems are mel frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980) and perceptual linear prediction (PLP) cepstral coefficients (Hermansky, 1990). In spirit both these features are similar as both use the knowledge about human speech perception. Even their extraction mechanisms are similar. The main steps to extract these features with in an analysis window are (Gold and Morgan, 2000):

1. Estimation of the power spectrum. In the case of MFCC extraction, the $\log$ power spectrum is estimated.

2. Integrating the power spectrum within overlapping critical band (Moore, 1997) filter responses to reduce the frequency sensitivity over original spectral estimate, particularly at high frequencies. In the case of MFCC, the "mel" filter banks (triangular in shape) are used to deform the frequency to follow the spatial relationship of hair cell distribution in the cochlea of the inner ear. The "mel" scale is based on speech perception. It uses a linear spacing upto 1 kHz and thereafter a logarithmic spacing (Moore, 1995). This spacing is related to the 24 "critical bands" of hearing research, although the use of 20 triangular filters are common in ASR. For the case of PLP, trapezoidally shaped filters are applied at roughly 1-Bark intervals. The Bark axis is derived by using a warping function (Hermansky, 1990).

3. Pre-emphasis of the spectrum to approximate the unequal sensitivity of human ear at different frequencies. In the case of MFCC, the preemphasis is done before $\log$ power spectrum estimation. For PLP, this done by weighting the elements of the critical band spectrum.

4. Compression of the spectral amplitudes. This done in PLP feature extraction by taking the cubic root which is an approximation to the power-law relationship between intensity and loudness (Stevens, 1957). In case of MFCC extraction nothing needs to be done here.

5. Performing inverse discrete Fourier transform (IDFT). In the case of MFCC, this step yields the cepstral coefficients. For PLP, the results are more like autocorrelation coefficients.

6. Performing spectral smoothing in order to reduce the effect of nonlinguistic variation (as described earlier in the previous section). In the case of MFCC, this is achieved by cepstral truncation i.e., while performing IDFT only lower 12 or 14 components are computed. For PLP feature extraction, an autoregressive model derived using the autocorrelation coefficients (Makhoul, 1975) obtained in the previous step is used to smooth the compressed critical band spectrum.

7. Orthogonal representation of the features. In the case of MFCC, orthogonalization is already done while performing IDFT. For PLP, the autoregressive coefficients (of the autoregressive model) are converted to cepstral coefficients through a simple recursion (Markel and Gray, 1976, Page 130). The orthogonal representation of the features help while modelling the distribution of the features in the later stage of ASR. In other words, if the features are orthogonalized and their distribution is modelled by a Gaussian then only the mean of the Gaussian and the diagonal covariance matrix (instead of full covariance matrix) needs to be estimated.

The principal difference between the extraction of MFCC feature and PLP feature is the nature of spectral smoothing. In case of MFCCs, it is cepstral based while in case of PLP it is linear prediction based.

Standard ASR systems treat each frame independently of the other (i.e., each frame of the input signal is analyzed separately). This is done in order to simplify computation. However, there is evidence that strong correlations exist across longer time spans due to coarticulation (Hermansky, 2003). In order to take into account the time correlation, the first order temporal derivatives (represented as Δs) and second order temporal derivatives ($\Delta\Delta$s) of the acoustic vectors are commonly used as additional acoustic parameters (Furui, 1981, 1986). Thus, in a standard ASR system the feature may consist of 13 cepstral coefficients, 13 Δs and 13 $\Delta\Delta$s, forming a 39 dimensional vector at each time frame. The approximate time derivatives can be estimated as proposed in (Furui, 1986). The frequency response of Δ and $\Delta\Delta$ filters according to (Furui, 1986) are centered at 15 Hz (for window of size 7). Recently, approaches have been proposed where the dynamic features are obtained by projecting the cepstral trajectories on sine and cosine basis functions (Kanedera *et al.*, 1998; Tyagi *et al.*, 2003). Yet another approach involves linear discriminant analysis (LDA) (e.g., Duda *et al.* (2001)). In this approach, a number of consecutive frames are concatenated to form high a dimensional vector ($13 \times 9 = 117$) and LDA is performed to identify the most relevant new dimensions (Haeb-Umbach and Ney, 1992). More recently, the time frequency correlation (Hermansky, 1999) has been exploited with standard frame-based ASR, such as, modulation spectrogram (Kingsbury *et al.*, 1998), TRAPS (Hermansky and Sharma, 1998), PLP^2 (Athineos *et al.*, 2004), spectro-temporal activity pattern (Ikbal *et al.*, 2004a). The recently proposed TANDEM approach derives a vector of posterior probabilities of subword units for every analysis frame from the input evidence, processes the posterior features to extract what we refer to as tandem features, which are then modelled by conventional acoustic modelling technique (Hermansky *et al.*, 2000). The TANDEM approach is described in detail later in Section 3.6. TRAP-TANDEM approach motivated by human auditory processing derives multiple evidence from a relatively long (500-1000 ms) and frequency-localized (1-3 Bark) overlapping time-frequency regions of the speech signal (TRAP) and, converts these evidences to features (TANDEM) (Hermansky, 2003).

The ASR problem would be greatly simplified, if the features have similar values for different repetitions of the same sound and, have distinct values for sounds that differ (w.r.t the semantic task of ASR). In this way, the sounds are well separated in the feature space. Due to the various reasons described earlier in Chapter 1, the acoustic features x_n exhibit considerable variability.

These variability lead to high variance with in the feature space of a sound. In order to handle this variability, in standard ASR systems, the features are post processed (before statistical inference (2.2)) for e.g., cepstral mean subtraction (Atal, 1974) or utterance/speaker level mean and variance normalization or filtering time trajectories of features such as RASTA filtering (Hermansky and Morgan, 1994), or processing during feature extraction such as vocal tract length normalization, usually implemented as warping of the spectrum during feature extraction (Andreou *et al.*, 1994; Lee and Rose, 1996).

Auxiliary Features

In the present work, we study different ways to use alternate features, such as, voice source characteristics (pitch frequency), rate-of-speech in order to improve the performance of the ASR system. We refer to these alternate features as "auxiliary features". Auxiliary features bring additional information, complementary to the usual features which can be integrated in standard ASR to reduce variability. We use auxiliary features that are estimated directly from the speech signal, namely, pitch frequency, short-term energy and rate-of-speech. They are all fundamental features of speech which change within a given speaker or/and utterance according to prosodic conditions and the environment. In Chapters 4 and 5, we present ASR studies using auxiliary features.

2.3 Acoustic Modelling

In the previous section, we described the transformation of digital signal into a sequence of acoustic features $X = \{x_1, \cdots x_n, \cdots x_N\}$. Since in most of the ASR systems the frame rate is uniform (i.e. frame shift P is the same throughout the utterance) the number of feature vectors N for a word may vary depending upon on the pronunciation and the speed at which it is pronounced. This leads to a paradigm where an arbitrary length continuous-valued acoustic feature sequence have to be matched with an arbitrary length sequence of discrete events of states (e.g., words).

In the early phase of ASR research, dynamic time warping (DTW) was widely used to address the above paradigm (Velichko and Zagoruyko, 1970; Sakoe and Chiba, 1978; Bridle *et al.*, 1983; Godin and Lockwood, 1989). In the DTW-based approach, one or more templates represent each word, where the template is the acoustic feature sequence X. The estimation and storage of the templates (reference template) comprises the training phase. During recognition, the test acoustic feature sequence (test template) is obtained and a global distance measure is accumulated as a sum of local distances, via successive comparisons between corresponding frames in the reference and test templates. In order to take into account the speaker variabilities, each of the words is compared with more than one reference template. The result of the DTW is the accumulated distance and, the warping path that minimizes the accumulated distance. The word sequence is then obtained from the warping path. DTW has large storage requirements as several templates of each word have to be stored. Warping constraints and the use of an appropriate distance measure attempts to account for variabilities at the cost of more computation. Nevertheless, in mobile phones DTW has been successfully used for name recognition (10-20 names).

State-of-the-art ASR systems use finite state machines (FSMs) to capture the great variability in the speech signal via stochastic modelling. The hope is to obtain good generalization without requiring storage of large amount of data (as opposed to DTW). The Markov model is an example of FSMs. Markov model is built up from a set of states $\mathcal{Q} = \{1, \cdots k, \cdots K\}$. The way these states are interconnected defines the topology of the Markov model. The topology of the Markov model helps to incorporate easily knowledge about lexical, syntactic, and semantic constraints. For instance, transition between states are allowed only if the resulting sequence of the states $Q = \{q_1, \cdots q_n, \cdots q_N\}$ with $q_n \in \mathcal{Q}$ produces a legal sentence following the system's grammar. The probability of a particular sequence of states Q is obtained by multiplying the state transition probabilities.

$$P(Q) = P(q_1) \prod_{n=2}^{N} P(q_n|q_1, \cdots q_{n-1}) \approx P(q_1) \prod_{n=2}^{N} P(q_n|q_{n-1}) \tag{2.7}$$

where, the latter part of the above equation results from the first order Markov assumption (i.e., the transition to any state at time $n+1$ depends only upon the state at time n). First order Markov models form the basis for hidden Markov models (HMMs) which are used in state-of-the-art ASR systems (Baker, 1975; Jelinek, 1976; Bahl *et al.*, 1983). Unlike the Markov model, in which each state corresponds to an observable event, in HMMs the observation is a probabilistic function of the state, resulting in a doubly embedded stochastic process model with an underlying stochastic process that is not observable, but only be observed through another set of stochastic processes that produce the sequence of observations (Rabiner, 1989). In other words, the details of the model's operation must be inferred through observations of speech (acoustic feature sequence), not from any internal representation such as, vocal tract positions or states. HMMs provide both segmentation and probability estimation capabilities. The main advantages of HMM-based ASR systems are (Baker, 1975; Jelinek, 1976; Makhoul and Schwartz, 1985; Levinson, 1985; Holmes and Huckvale, 1994):

- Architecture: The states in the HMM capture short-term spectral characteristics and, the temporal relationship through Markov chain. The use of probabilities to express the output distribution of models allows the models to generalize easily to unseen data. The HMM-based approach provides a tractable mathematical structure.
- Training: The parameters of the HMM-based ASR system can be trained using a large amount of training data and little speech knowledge.
- Flexibility: The HMM-based ASR system allows flexibilities, such as,
  - Choosing a method of acoustic analysis. For instance, MFCC or PLP.
  - Applying speech knowledge through specifying structure of the models themselves. For example, the word models can be created by concatenation of subword units such as, phoneme in a particular sequence.
  - It can handle duration, spectral complexity and variability of the sounds being produced through model topology. For instance, minimum duration constraint on the HMM states or the use of context-dependent subword units.

If Q is an HMM state sequence in Markov model W and X is the observation sequence then the probability of that sequence is given by:

$$p(X|W) = \sum_{Q} p(X|Q, W)P(Q|W) \tag{2.8}$$

where $\sum_Q$ implies summation over all possible state sequences Q in W. The probability of the state sequence $P(Q|W)$ can be computed through (2.7) and the $p(X|Q, W)$ can be estimated as:

$$p(X|Q, W) = \prod_{n=1}^{N} p(x_n|x_1, \cdots x_{n-1}, Q, W) \tag{2.9}$$

In order to simplify the computation, an important assumption is made (c.i.i.d assumption): the probability of the observed symbol at a certain time n i.e., x_n only depends on the current state q_n and, is conditionally independent of any other observations given the state q_n, and they are identically distributed, resulting in:

$$p(X|Q, W) = \prod_{n=1}^{N} p(x_n|q_n) \tag{2.10}$$

The value of the probability density function $p(X|W)$ for a certain observation sequence X is called the likelihood of X. Every state has two probability distribution, namely, state-transition probabilities a_{ij}:

$$a_{ij} = P(q_n = j|q_{n-1} = i) \tag{2.11}$$

and state emission probability density function $b_j(x_n)$:

$$b_j(x_n) = p(x_n|q_n = j) \tag{2.12}$$

The overall likelihood $p(X|W)$ can then be rewritten as:

$$p(X|W) = \sum_Q (\prod_{n=1}^{N} p(x_n|q_n)P(q_n|q_{n-1}, W)) \tag{2.13}$$

The complete definition of the HMMs consists of the HMM topology, for each state the emission PDF and state transition probabilities. The parameters Θ_a of the acoustic model in (2.2), are the parameters of the emission PDFs $b_j(x_n)$ and the state transition probabilities a_{ij} and, these parameters are trained from a training data. The most common training algorithm is Baum-Welch algorithm, also called forward-backward algorithm (Baum *et al.*, 1970). This algorithm is a specific case of expectation-maximization (EM) algorithm (Dempster *et al.*, 1977). The HMMs can also be trained using embedded Viterbi training algorithm which is an approximation of the forward-backward training algorithm (Merhav and Ephraim, 1991b; Viterbi, 1967; Rabiner and Juang, 1993; Morgan and Bourlard, 1995; Gold and Morgan, 2000). In Chapter 3, we describe these two algorithms in detail. The HMM training is an iterative procedure, where the training starts with an initial estimate of the parameters, then an iterative reestimation of the parameters of the HMMs is done in such way that they yield better models. We discuss about the training procedure in more detail in the next chapter.

As described earlier, the acoustic feature vectors exhibit variability due to different factors. Some of the approaches try to take care of these variations at feature level. However, due to problem of unknown speech distribution, sparse training data and feature level variabilities (both spectral and temporal), and mismatch between training and testing conditions, the system generally incorporates a small amount of speaker and environment specific adaptation data into the training process. The most common adaptation techniques are maximum likelihood linear regression (MLLR) (Leggetter and Woodland, 1995) and maximum a posteriori (MAP) adaptation (Gauvain and Lee, 1994). For a good review about speaker adaptation methods for automatic speech recognition refer to (Woodland, 2001).

Standard HMM-based ASR systems usually process speech with frame updates of 10 ms. As we saw earlier in this section, HMMs make first-order assumption and assume that the emission of an acoustic observation x_n at time n is conditionally independent of the previous states and acoustic observations given the present state q_n, and the acoustic observations are identically distributed. This assumption simplifies the training and testing procedure, but leads to ignorance of past history and subsequently forces the use of less precise PDFs (i.e., with large variances). In recent studies, attempts have been made to exploit the timing information, such as stochastic trajectory models (Ostendorf *et al.*, 1996) and trajectory modelling (Tokuda *et al.*, 2003).

Modelling Auxiliary Sources of Knowledge in Standard ASR

The central theme of the present work is to integrate auxiliary knowledge sources so as to make the acoustic models more robust to the variabilities present in the speech signal. The variabilities present in the speech signal can be reduced by integrating (a) auxiliary features and (b) auxiliary subword units. In the present work, we integrate auxiliary knowledge source $A = \{a_1, \cdots a_n, \cdots a_N\}$ in standard HMM-based ASR system by modelling the joint likelihood $p(Q, X, A)$:

1. We may use the auxiliary feature sequence A as:

    (a) An additional feature, i.e., estimating emission distribution as $p(y_n|q_n)$, where, y_n is augmented feature vector $y_n = (x_n, a_n)$.

    (b) A variable conditioning the emission distribution, i.e., estimating the emission distribution as $p(x_n|q_n, a_n)$.

    We present the different ways to integrate auxiliary features in standard HMM-based ASR in Chapter 4.

2. When modelling auxiliary subword unit sequence A, there are two chains of subword units corresponding to state sequence Q and state sequence A. The emission distribution is estimated as $p(x_n|q_n, a_n)$, where a_n is now a discrete HMM state. We study the modelling of auxiliary subword units in standard HMM-based ASR system in Chapter 7.

In ASR systems, the HMMs could be the reference models of words or units smaller than words. In the following section, we describe what does the HMMs represent in standard HMM-based ASR system.

2.4 Subword Units and Pronunciation Modelling

In the early days of HMM-based ASR, word models were commonly used. Nowadays, it is more common to find the state-of-the-art ASR systems using HMMs to model subworod units such as, phonemes. Training of word HMMs puts more onus on the training data collection as sufficient instances of all the words has to be collected in order to get a good estimate of the HMM parameters. Still, in tasks such as connected digit recognition where, the number of words to be recognized is just eleven, word models may still be trained. In the case of large vocabulary systems, this is not feasible, as it is possible to observe words during recognition that were not seen during training. In later HMM-based ASR studies, the use of HMMs for smaller phonological recognition units (subword units) such as, phonemes became more common (e.g., Gold and Morgan, 2000, Chapter 23). The subword unit HMMs are more general and, they can be connected together to form word HMMs. For instance, the HMMs of phonemes /ey/ and /t/ can be connected in left-to-right topology to form word model of word *eight*. The lexicon of the ASR contains the words and their transcription in terms of subword units (pronunciation).

In state-of-the-art ASR systems, the phoneme is commonly used as the subword unit. The use of phonemes as subword units is motivated by linguistic studies (e.g., Gold and Morgan, 2000, Chapter 23). Phonemes can be defined as the smallest unit of sound in a spoken language i.e., smallest unit that defines lexical contrast. In other words, a single distinguishable sound within a particular language. In English language, there are around 42 phonemes[4]. The phonetic pronunciation of the words can be obtained from a standard lexical dictionary. Standard ASR systems train phoneme models and then connect them according to the pronunciation in "beads-in-a-string" fashion (Ostendorf, 1999) to create word models. The phoneme models can be context-independent (CI) i.e., each phoneme model is trained independent of others. The advantage of using CI phonemes is that there are fewer models to be trained, but they fail to model the coarticulation and stress which can extend well beyond a single phoneme. In order to model coarticulation in standard ASR, context-dependent (CD) (typically with one preceding and one succeeding context) phoneme models are trained (Schwartz *et al.*, 1985; Lee, 1990). If there are K phonemes, there are K^3 possible context-dependent models (not all combinations are allowed). In order to train all of these models properly large quantity of training data is required. Also, having a separate CD HMM for each sequence of three phonemes is inefficient, because many contexts have very similar coarticulatory effects. Hence, in state-of-the-art ASR tied models are used (Young, 1992; Ljolje, 1994). Tied models

[4] The exact number of phonemes depends upon accent and dialect.

share the same parameter values across different models, greatly reducing the number of parameters that have to be trained. Tying can be automatic (e.g., data-driven decision trees) or guided by linguistic properties (Odell, 1995). CD phonemes can capture most local coarticulations, though to further model the coarticulation more context is used (≥ 2 phonemeic context[5]) which again raises the issue of identifying and training them. Research efforts are also in the complementary direction of capturing the coarticulation effect through the acoustic feature sequence, rather than through explicit modelling (Hermansky, 2003).

As mentioned earlier, the lexicon contains all the words and their phonetic transcription. In speech recognition it is typically assumed that the speakers pronounce the words exactly according to the pronunciation given in the lexicon. This may hold true for read or carefully articulated speech, but in case of spontaneous or natural speech speakers exhibit pronunciation variation. These variations can occur at the acoustic level due to change of rate-of-speech, speaking style, different accents, different length of vocal tract stress, emotion or can manifest itself at the lexical level due to phonological processes such as, assimilation, coarticulation, deletion, insertion, substitution (Strik and Cucchiarini, 1999). At the acoustic level, the pronunciation variation can be modelled by iterative training and the sharing of model parameters (Sarclar, 2000). At the lexical level, the pronunciation variation may be modelled by extracting multiple pronunciations of the words and including them in the lexicon. The approach for multiple pronunciation extraction can be data-driven, knowledge-based or a mix of both (Strik and Cucchiarini, 1999).

The use of the phonemes as subword units is a notion borrowed from linguistic studies (e.g., Gold and Morgan, 2000, Chapter 23). In the case of ASR, phoneme may not be the best subword unit. However, ASR research questioning this has focussed on the use of subword units such as, syllable (Ganapathiraju *et al.*, 2001) and automatically-derived subword units (Bacchiani and Ostendorf, 1999; Singh *et al.*, 2002).

Pronunciation Model Evaluation

In HMM-based ASR, usually every word is modelled by a sequence of subword units (usually phonemes), also called baseform pronunciation model. The sequence of subword units in the baseform pronunciation model are nothing but lexical constraints. We propose an approach to evaluate the adequacy of the baseform pronunciation model of words by perturbing (relaxing) the lexical constraints, inferring new pronunciation variants and, assessing the "stability" of the pronunciation model by evaluating the inferred pronunciation variants through a combination of reliability and proximity measures. The reliability measure tells how good is the match between the acoustic observation X and the inferred pronunciation variant. The proximity measure tells how close is the inferred pronunciation model compared to the baseform pronunciation model. If the pronunciation model is stable, then, the reliability should be "high" and proximity should be "close". Through this approach, we show that integrating auxiliary features in standard ASR system improves the matching and discriminative properties of the baseform pronunciation model. The proposed approach to evaluate baseform pronunciation model also allows to generate new pronunciation variants which when added to the lexicon improves the performance of ASR. Chapter 6 presents the proposed approach to evaluate baseform pronunciation model.

Auxiliary Subword Units

As described earlier in this section, standard ASR systems usually use phonemes as subword units. In the present work, we use graphemes as auxiliary subword units. The main advantages of using graphemes as subword units is that (a) definition of the lexicon is easy, (b) the pronunciation model is unique (only one way to write a word), and (c) graphemes may carry information that is complimentary to phonemes, thus may help in improving the match between the observation and pronunciation model. Here, every word pronunciation model is defined in terms of phoneme units

[5] For instance, using two preceding and two following phonemes.

and grapheme units. During training, acoustic models are jointly trained for both the subword units. During recognition, either both or one of the subword unit representation is used. We refer to this system as phoneme-grapheme based ASR system. We present the phoneme-grapheme based ASR system in Chapter 7.

2.5 Language Modelling

In the beginning of this chapter, the basic paradigm of statistical speech recognition was defined in (2.2). In continuous speech recognition, W is a sequence of words of unknown length. The language model provides an estimate of the probability $P(W|\Theta_l)$. The use of such prior probability is not only used in speech recognition, but also in other sequence processing tasks such as translation and optical character recognition. The language model usually is a Markov model. For a given sequence of L words, the language model probability can be estimated as:

$$P(W_1^L) = \prod_{l=1}^{L} P(w_l|w_{l-1}, \cdots w_1), \tag{2.14}$$

where, W_1^L is a sequence of words $w_1, \cdots w_l, \cdots w_L$. This means that, if the probability of a certain sentence (word sequence W_1^L) is known, the probability of a sentence word with one extra word (w_{L+1}) can be computed efficiently by simple multiplication. The conditional probabilities can be directly estimated from text data, but in spontaneous speech the sentences are not always grammatically correct. To account for this, word transcriptions of the ASR speech training data are also used. If the vocabulary is small, and if there is sufficient training data to estimate the conditional probabilities $P(W_l|w_{l-1}, \cdots w_1)$, then the task of estimating $P(W_1^L)$ is also made easier. However, for large vocabulary speech recognition, the size of the lexicon is typically 50000 words or more. In such cases, it is difficult to explicitly store the estimates for all potential word sequences. A solution to this problem is clustering a set of possible word histories. The easiest and most commonly used method for clustering word histories is simple truncation of the word history, after a certain number of words, N. In the literature, this is generally referred to as an N-gram model (Bahl *et al.*, 1983; Nádas, 1984). The N-gram estimates the probability of each word in the context of its preceding N $-$ 1 words. Thus, bigram models (N $=$ 2) use statistics of word pairs and trigrams (N $=$ 3) model word triplets. The unigram probability is the prior probability of the word. In a trigram model, the conditional probabilities are estimated as:

$$\hat{P}(w_l|w_{l-1}, w_{l-2}) = \frac{N(w_l, w_{l-1}, w_{l-2})}{\sum_w N(w, w_{l-1}, w_{l-2})}, \tag{2.15}$$

where $N(\cdot)$ is frequency count of word triplets. This conditional probability is then used to compute the probability of the word sequence. The estimation of $\hat{P}(w_l|w_{l-1}, w_{l-2})$ is easy, if the size of the vocabulary is small, but for large vocabulary sizes ≥ 50000 huge training (text) data will be required to estimate as 50000^3 (not all word combination are allowed) conditional probabilities will have to be estimated. In practice, bigrams or unigrams are employed, and trigrams are used if there are sufficient training resources. However large the training data may be there will be certain word sequences that do not occur in the training, but they may occur during recognition. Consequently, during N-gram LM estimation a certain amount of probability mass is allocated to the unseen N-grams.This method is called discounting. Yet another way to estimate the probability of unseen N-grams are the back-off methods that rely on lower order statistics (i.e., use N $-$ 1-grams for unseen N-grams) (Katz, 1987). Interpolated LM combines the different N-gram statistics by a weighted sum as shown below:

$$\alpha_0 + \alpha_1 P(w_l) + \alpha_2 P(w_l|w_{l-1}) + \alpha_3 P(w_l|w_{l-1}, w_{l-2}),$$

where, α_i are i-gram weights and they sum to one. Suppose that a certain trigram is unseen then the weight α_3 is assigned a value of zero and other weights are elevated.

2.6 Decoding

The last component of the automatic speech recognition system is the decoder. The decoder combines the acoustic model likelihood and language model probabilities to output the word sequence. Decoding in statistical speech recognition involves a search for the best possible word sequence $\hat{W}$ given acoustic observation sequence X:

$$\hat{W} = \arg\max_{W} p(X|W, \Theta_a) P(W|\Theta_l)$$

where, $\hat{W}$ is the word sequence from the set of all possible word sequences that has the highest posterior probability given the acoustic model and language model and, Θ_a and Θ_l are the parameters of acoustic model and language model, respectively[6]. Using (2.8) this could be expanded as:

$$\hat{W} = \arg\max_{W} P(W) \sum_{Q} p(X|Q, W) P(Q|W) \tag{2.16}$$

The evaluation of the above equation is computationally expensive as the number of words in sentences increases, so is the sum over all possible state sequences. In order to avoid this computation, the sum operation is approximated by the max operation, as shown below:

$$\hat{W} = \arg\max_{W} P(W) \ \arg\max_{Q} p(X|Q, W) P(Q|W) \tag{2.17}$$

This best path search through all possible state sequences is called Viterbi decoding (Viterbi, 1967; Forney, 1973). The traditional way to find the best path is to examine all possibilities before rejecting any (Ney, 1984; Ney and Ortmanns, 2000). For a large vocabulary ASR system, this still demands large computational resources and time. In the literature, more efficient search methods have been proposed such as, beam-search (Lowerre, 1976; Klatt, 1977) and stack decoding (Jelinek, 1969; Bahl *et al.*, 1983).

In beam-search, the decoder examines a narrow beam of likely alternatives around the locally best path. This is done by imposing a beam width δ around the probability p of the most likely partial hypothesis at that time and removing (pruning) all hypotheses that have probability below $p - \delta$. The pruning of the hypotheses can also done by phone look-ahead technique, where each time a hypotheses crosses the phone boundary, the decoder examines the next few frames to see if they give sufficiently high likelihoods. Yet another approach is the multipass strategy, which uses a first pass in which a coarse recognition is done using simple models that are less costly in memory and time while retaining only the N-best hypotheses. This reduces the number of paths to be considered for the second pass during which a detailed analysis is done. In the second pass, the N-best hypotheses are rescored and the best hypothesis is selected. Though, such approaches make the search faster by avoiding examination of many useless paths, they can also discard some correct hypotheses prematurely.

The acoustic model likelihoods and language model probabilities have different dynamic ranges. The acoustic model likelihood is a joint likelihood of N (number of frames) transition probabilities and N emission probabilities (low values), whereas, the language model probabilities are simply a function of N-gram probabilities (high values). Hence, during decoding the acoustic model probabilities may dictate the choice of best hypotheses. In order to overcome this problem, the $\log$[7] language model probabilities are scaled by a certain factor. This is called language scaling factor. This scaling factor is empirically determined to maximize the ASR accuracy (Jelinek, 1976; Bahl *et al.*, 1980; Takeda *et al.*, 1998).

[6] For simplicity, Θ_a and Θ_b will be dropped in rest of this section.

[7] The computations are done with log probabilities in order to avoid underflow and minimize the number of multiplications.

During recognition, most of the errors stem from insertion of words with a small number of phonemes. This happens because the low acoustic cost is combined with high probability of occurrence for a wide range of contexts. To alleviate this problem, a word insertion penalty is added during decoding to penalize a higher numbers of words in a sentence (Jelinek, 1976; Bahl *et al.*, 1980; Takeda *et al.*, 1998).

2.7 Summary

In this chapter, we described the different components of an automatic speech recognition system. This is also summarized in the Figure 2.1. We also briefly introduced the main contributions of the present work with respect to different components of ASR system.

In the following chapter, we describe in detail the acoustic modelling process in HMM-based ASR system and, describe the different HMM-based ASR systems that are being used in the present work.

Chapter 3

HMM-Based ASR Systems and Experimental Setup

3.1 Introduction

HMM-based ASR systems have to tackle three basic problems (Rabiner, 1989):

1. Estimation: Given the observation sequence X, the Markov Model W, and the acoustic model parameters Θ_a, how to estimate efficiently the likelihood of the data $p(X|\Theta_a, W)$?

2. Decoding: Given the observation sequence X, the Markov Model W, and the acoustic model parameters Θ_a, how to find the optimal state sequence that maximizes the likelihood of the data $\max_Q p(Q, X|\Theta_a, W)$?

3. Training: How to estimate parameters Θ_a of the acoustic model so as to maximize the likelihood $p(X|\Theta_a)$ of the training data X?

In Section 3.2, we describe two different estimates the likelihood $p(X|\Theta_a, W)$ of the data X, namely, full likelihood and Viterbi likelihood estimation, and briefly describe the solution to the problem of decoding which is the by product of the Viterbi likelihood estimation. Section 3.3 describes the training of HMMs, i.e., estimation of the acoustic model parameters Θ_a. Sections 3.4 and 3.5 describe the state-of-the-art HMM/Gaussian mixture models (HMM/GMM) systems and hybrid HMM/ANN systems, respectively. Sections 3.6 and 3.7 describe briefly the recently proposed TANDEM systems and dynamic Bayesian Networks-based systems for ASR, respectively. Section 3.8 briefly discusses about the different ASR tasks that have been addressed in this thesis, their experimental setup and the metrics used to evaluate the performance of the ASR systems.

3.2 Estimation and Decoding

Given the HMM model parameters Θ_a and the Markov model W (e.g., pronunciation model of a word), the likelihood of the data $p(X|\Theta_a, W)$ can be estimated in the following two ways:

- Full likelihood: The full likelihood is the likelihood of all possible paths Q in W. It can be estimated by the forward pass of Baum-Welch algorithm,

$$p(X|\Theta_a, W) = \sum_Q p(Q, X|\Theta_a, W) \tag{3.1}$$

- Viterbi likelihood: The Viterbi likelihood is the likelihood of the best path Q^* in W. It is estimated by replacing the *sum* operation in (3.1) by *max* operation as shown below,

$$p(Q^*, X|\Theta_a, W) = \max_Q p(Q, X|\Theta_a, W) \tag{3.2}$$

3.2.1 Full Likelihood

The forward likelihood in Baum-Welch algorithm is defined as:

$$\alpha(n, j) = p(X_1^n, q_n = j|\Theta_a, W)$$

where, $\alpha(n, j)$ is the joint likelihood of being in state j at time n having observed acoustic sequence $X_1^n = x_1, \cdots x_n$. If $b_j(x_n) = p(x_n|q_n = j)$ is the likelihood of the acoustic vector x_n being emitted by state j and, $a_{ij} = p(q_n = j|q_n = i)$ is the transition probability from state i to state j and, $\pi(j)$ is the probability that the state sequence starts with state j, then the forward likelihoods can be computed recursively (also called α recursion or forward pass):

1. Initialization:

$$\alpha(1, j) = \pi(j)b_j(x_1), \qquad 1 \leq j \leq K$$

2. Recursion for $2 \leq n \leq N$

$$\alpha(n, j) = \left(\sum_{i=1}^{K} \alpha(n-1, i)a_{ij}\right) b_j(x_n), \qquad 1 \leq j \leq K$$

3. Termination:

$$p(X|\Theta_a, W) = \sum_{i=1}^{K} \alpha(N, i).$$

The result of the forward pass is the likelihood $p(X|\Theta_a, W)$.

3.2.2 Viterbi Likelihood

The Viterbi likelihood is defined as

$$V(n, j) = \max_{q_1, \cdots, q_{n-1}} p(X_1^n, q_1, \cdots q_{n-1}, q_n = j|\Theta_a, W)$$

where, $V(n, j)$ is the joint likelihood of the best path (among the all possible paths) being in state j at time frame n having observed acoustic sequence X_1^n. If $\Psi_j(n)$ is the state q_n^* visited by the best path at time frame n, the Viterbi likelihood for the whole utterance can be estimated (using the parameters Θ_a) as:

1. Initialization:

$$\begin{aligned} V(1, j) &= \pi_j b_j(x_1), \qquad 1 \leq j \leq K \\ \Psi_j(1) &= 0 \end{aligned}$$

2. Recursion for $2 \leq n \leq N$

$$\begin{aligned} V(n, j) &= \left(\max_{1 \leq i \leq K} V(n-1, i)a_{ij}\right) b_j(x_n), \qquad 1 \leq j \leq K \\ \Psi_j(n) &= \arg\max_{1 \leq i \leq K} V(n-1, i)a_{ij}, \qquad 1 \leq j \leq K \end{aligned}$$

3. Termination:

$$\begin{aligned} p(Q^*, X|\Theta_a, W) &= \max_{1\le i\le K} V(N,i) \\ q_N^* &= \arg\max_{1\le i\le K} V(N,i). \end{aligned}$$

4. Backtracking (optimal state sequence):

$$q_n^* = \Psi_{q_{n+1}^*}(n+1), \qquad N-1 \ge n \ge 1.$$

This algorithm is referred to as Viterbi algorithm (Viterbi, 1967; Forney, 1973). The result of the Viterbi algorithm is the joint likelihood $p(Q^*, X|\Theta_a, W)$ and the optimal state sequence $Q^* = \{q_1^*, \cdots q_n^*, \cdots q_N^*\}$. The optimal state sequence is also referred to as best path or segmentation. Implementation wise except for backtracking the major difference between forward pass and Viterbi algorithm is that the *sum* operation in forward pass is replaced by the $\max$ operation in Viterbi algorithm.

The Viterbi algorithm is also the solution for the aforementioned second basic problem in HMM-based ASR, decoding, as the optimal sequence is one of the outcome of Viterbi algorithm. The Viterbi algorithm described in this subsection shows how we can obtain the optimal state sequence. The same Viterbi algorithm can be extended to obtain the word sequence (output of the ASR system). Section 2.6 in the previous chapter briefly describes how the word sequences are obtained based on the Viterbi algorithm.

3.3 Training

In the previous section, we assumed that the estimates of the parameters Θ_a of the HMM system, i.e., the initial distribution π, transition probabilities a_{ij}, and parameters of emission distribution $b_j(x_n)$ for all the states are available. One of the difficult task in HMM-based ASR system is to obtain a reliable estimate of these parameters from the training data. This is the aforementioned third basic problem of HMM-based ASR system.

The training of the HMM-based ASR will be greatly simplified if the segmentation of the training data is available in terms of the states of the HMM. In other words, for every acoustic feature vector x_n there is a true state identity associated with it. If the training set consists of B utterances and, the number of frames corresponding to each of the utterance is $N_1, \cdots N_b, \cdots N_B$, respectively, then the transition probability from any HMM state i to any HMM state j can be simply estimated as:

$$a_{ij} = \frac{\sum_{b=1}^{B}\sum_{n=1}^{N_b-1} P(q_{n+1}=j|q_n=i,b)}{\sum_{b=1}^{B}\sum_{n=1}^{N_b} P(q_n=i|b)} \tag{3.3}$$

where, $P(q_n = i|b)$ is the probability of being in state i at time frame n of utterance b and $P(q_{n+1} = j|q_n = i, b)$ is the probability of being in state i at time frame $n-1$ and in state j in time frame n of utterance b. Given the segmentation of the training data $P(q_n = i|b)$ and $P(q_{n+1} = j|q_n = i, b)$ are either 0 or 1.

The initial distribution can be estimated as:

$$\pi_j = \frac{\sum_{b=1}^{B} P(q_1 = j|b)}{B} \tag{3.4}$$

where, $P(q_1 = j|b)$ is 0 or 1 given the segmentation.

Suppose if the emission distribution of the each state is modelled by single Gaussian, then parameters of the emission distribution mean μ and covariance Σ, for any HMM state j can be estimated as:

$$\mu_j = \frac{\sum_{b=1}^{B}\sum_{n=1}^{N_b} P(q_n = j|b)\cdot x_n^b}{\sum_{b=1}^{B}\sum_{n=1}^{N_b} P(q_n = j|b)} \tag{3.5}$$

$$\Sigma_j = \frac{\sum_{b=1}^{B}\sum_{n=1}^{N_b} P(q_n = j|b)\cdot(\mu_j - x_n^b)(\mu_j - x_n^b)^T}{\sum_{b=1}^{B}\sum_{n=1}^{N_b} P(q_n = j|b)} \tag{3.6}$$

where, $X_b = \{x_1^b, \cdots x_n^b, \cdots x_{N_b}^b\}$ of is the acoustic observation sequence of training utterance b. Again, given the segmentation $P(q_n = j|b)$ is 0 or 1.

Since the states in the HMM are hidden, there is no closed-form equation for estimating the parameters. This leaves us with the problem of estimating the posteriors $P(q_n = i|b)$ and $P(q_{n+1} = j, q_n = i|b)$ from the training data (acoustic observation X_b).

The HMM is usually trained in the maximum likelihood framework where, it is assumed that there is a single critical set of parameters, but it is unknown. The training is based on the definition of an auxiliary function $Q(\hat{\Theta}_a, \Theta_a)$ of current parameter set Θ_a and the re-estimated parameter set $\hat{\Theta}_a$. The definition of this auxiliary function guarantees that the maximization of $Q(\hat{\Theta}_a, \Theta_a)$ leads to increased likelihood i.e., $p(X|\hat{\Theta}_a) \geq p(X|\Theta_a)$ (X is the complete training data), subsequently converging to a optimal set of parameters. The maximum likelihood framework of HMM training is a special case of expectation-maximization (EM) algorithm. The EM has two steps, namely, expectation (E-step) and maximization (M-step). In the E-step, the posteriors $P(q_n = i|b)$ and $P(q_{n+1} = j|q_n = i, b)$ are "estimated from the training data". In the M-step, the parameters Θ_a are re-estimated using $P(q_n = i|b)$ and $P(q_{n+1} = j|q_n = i, b)$ in Equations (3.3), (3.4), (3.5) and (3.6) yielding $\hat{\Theta}_a$.

In standard HMM-based ASR, in order to simplify the training typically a Markov model W_b for each training utterance b in terms of the HMM states is created. In other words, given the word level transcription of the training utterance b, the phoneme HMMs are concatenated to create word model based on the pronunciation model in the dictionary and the word models are concatenated to create the sentence model, ultimately yielding Markov model W_b. Given the acoustic observations of all the training utterances and their respective Markov models, the training of the HMM then involves the following:

1. Initialization of the parameters Θ_a. For instance, initializing the emission distribution with a zero mean and unit variance Gaussian.

2. E-step: Estimation of posteriors $P(q_n = i|X_b, \Theta_a, W_b)$ and $P(q_{n+1} = j|q_n = i, X_b, \Theta_a, W_b)$ for each utterance b in the training data, where

$$\gamma_b(n, j) = P(q_n = j|X_b, \Theta_a, W_b) \tag{3.7}$$

$$\xi_b(n, i, j) = P(q_{n+1} = j|q_n = i, X_b, \Theta_a, W_b) \tag{3.8}$$

   $\gamma_b(n, j)$ is the probability of being in state j at time instant n given X_b, W_b and Θ_a and, $\xi_k(n, i, j)$ is the probability of being in state i at time instant n and in state j at time instant $n+1$ given X_b, W_b and Θ_a

3. M-step: Re-estimation of the parameters yielding $\hat{\Theta}_a$, by replacing $P(q_n = j|b)$ and $P(q_{n+1} = j|q_n, b)$ in Equations (3.3), (3.4), (3.5) and (3.6) by $\gamma_b(n, j)$ and $\xi_b(n, i, j)$, respectively.

4. Evaluation of the auxiliary function $Q(\hat{\Theta}_a, \Theta_a)$. This done by estimating $p(X|\Theta_a)$ and $p(X|\hat{\Theta}_a)$. if $p(X|\hat{\Theta}_a) \geq p(X|\Theta_a)$ then replace Θ_a by new estimate of the parameters $\hat{\Theta}_a$ and go to Step 2, else terminate the training with Θ_a as the trained parameters.

The training is an iterative process, with each iteration consisting of one E-step and one M-Step.

As described above, in order to train the parameters of the HMMs, we need to estimate $\gamma_b(n,j)$ and $\xi_b(n,i,j)$ for each training utterance b. There are two ways to estimate them, namely, (1) forward-backward algorithm and (2) Viterbi algorithm. When $\gamma_b(n,j)$ and $\xi_b(n,i,j)$ are estimated using forward-backward algorithm, the training is generally referred to as Baum-Welch training (Baum *et al.*, 1970) or forward-backward training and, when they are estimated using Viterbi algorithm, the training is referred to as embedded Viterbi training.

3.3.1 Forward-Backward Algorithm

In the forward-backward algorithm, at every time frame n two joint likelihoods are estimated, namely, (1) forward likelihood $\alpha_b(n,j)$ and (2) backward likelihood $\beta_b(n,j)$. The forward likelihood $\alpha_b(n,j)$ is the joint likelihood of being in state j at time n having observed acoustic sequence $X_{1,b}^n = x_1^b, \cdots x_n^b$ of utterance b given the parameters Θ_a and Markov model W_b. In Section 3.2.1, we described how this forward likelihood can be estimated.

The backward likelihood $\beta_b(n,j)$ is the joint probability of being in state j at time frame n and observing $X_{n+1,b}^N$ of utterance b in time frames $n+1, \cdots N$ given the parameters Θ_a and Markov model W_b. The complete procedure to compute $\beta_b(n,j)$ for the training utterance b is as following:

1. Initialization

$$\beta_b(N_b, i) = 1, \qquad 1 \leq i \leq K.$$

2. Recursion $N_b - 1 \geq n \geq 1$

$$\beta_b(n,i) = \sum_{j=1}^{K} a_{i,j} b_j(x_{n+1}^b) \beta_b(n+1,j), \qquad 1 \leq i \leq K$$

3. Termination

$$p(X_b|\Theta_a, W_b) = \sum_{j=1}^{K} \beta_b(1,j).$$

At any given time n of utterance b in the forward-backward algorithm, the forward variable $\alpha_b(n,j)$ accounts for observations $X_{1,b}^n$ and the backward variable $\beta_b(n,j)$ accounts for observation $X_{n+1,b}^N$. Given $\alpha_b(n,j)$ and $\beta_b(n,j)$, we can estimate $\gamma_b(n,j)$ in the following way:

$$\gamma_b(n,j) = P(q_n = j|X_b, \Theta_a, W_b) = \frac{\alpha_b(n,j)\beta_b(n,j)}{\sum_{i=1}^{K} \alpha_b(n,i)\beta_b(n,i)} \tag{3.9}$$

Similarly, we can estimate $\xi_b(n,i,j)$ as:

$$\xi_b(n,i,j) = P(q_{n+1} = j|q_n = i, X_b, \Theta_a, W_b) = \frac{\alpha_b(n,i) a_{ij} b_j(x_{n+1}^b) \beta_b(n+1,j)}{\sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_b(n,k) a_{kl} b_l(x_{n+1}^b) \beta_b(n+1,l)} \tag{3.10}$$

3.3.2 Viterbi Algorithm

The $\gamma_b(n,j)$ and $\xi_b(n,i,j)$ for each training utterance b can be obtained by Viterbi algorithm in the following way:

1. Given X_b, W_b and Θ_a, run the Viterbi algorithm to obtain the segmentation (i.e., get optimal state sequence).

2. Given the segmentation, $\gamma_b(n, j)$ and $\xi_b(n, i, j)$ are either 0 or 1.

In Section 3.2.2, we have described the Viterbi algorithm.

The embedded Viterbi training is an approximation of Baum-Welch training algorithm where, the sum over all potential states is replaced by a maximization operation (Viterbi approximation). This makes the training faster as the computational cost of obtaining Viterbi segmentation is less than the forward-backward algorithm. Moreover, it has also been shown that the exact (Baum-Welch) and approximate (Viterbi) estimates are close with a sufficient number of frames (Merhav and Ephraim, 1991a).

The acoustic feature x_n can be discrete valued (quantization of feature space into codebooks) or continuous valued, consequently the emission distribution can be a discrete probability distribution or probability density function. State-of-the-art HMM-based ASR systems generally make use of continuous valued features. In the following Sections, we describe different HMM-based ASR which have been studied in this thesis. Based on the way the emission distribution is modelled, there are two different types of HMM-based ASR systems, namely: (a) HMM/GMMs ASR system and (b) hybrid HMM/ANN ASR systems. In the following section, we describe the HMM/GMM ASR system.

3.4 HMM/GMM ASR System

One of the main criteria in the selection of an appropriate output distribution is the availability of a scheme to estimate the parameters of the distribution. Liporace showed that a relatively broad class of elliptical symmetric distributions such as multivariate Gaussian probability density function satisfy this necessary criteria (Liporace, 1982). In the previous section, for simplicity we considered the emission distribution being modelled by a single Gaussian. As, we observed earlier in Chapter 2, the acoustic feature vectors x_n are spread out in the feature space due to different variabilities present in the speech signal. Some of these variabilities are reduced at the feature level, but the remaining variability has to be captured by the statistical model of the acoustic feature distribution. Hence, in practice the emission distribution is a mixture of distributions i.e., mixture of Gaussian densities:

$$b_j(x_n) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(x_n, \mu_{jm}, \Sigma_{jm}) = \sum_{m=1}^{M} c_{jm}\frac{1}{2\pi^{\frac{d}{2}}|\Sigma_{jm}|^{\frac{1}{2}}}\exp^{-\frac{1}{2}(x_n-\mu_{jm})^T\Sigma_{jm}^{-1}(x_n-\mu_{jm})} \tag{3.11}$$

where, M is the number of mixtures, c_{jm} is the mixture weight, μ_{jm} and Σ_{jm} are the mean vector and covariance matrix of m^{th} multivariate Gaussian density function of state j, respectively. $|\Sigma_{jm}|$ is the determinant of Σ_{jm} and, d is the dimension of the feature space. c_{jm} is positive and $\sum_{m=1}^{M} c_{jm} = 1.0$. Every emitting state in the HMM is modelled by a mixture of Gaussian densities and their parameters are estimated during training (as part of the Baum-Welch algorithm).

During training in the E-step, $\gamma(n, j, m)$ the posterior probability of belonging to m^{th} Gaussian of state j at time frame n is estimated using $\gamma(n, j)$ in (3.7) as (Rabiner and Juang, 1993; Bilmes, 1997)

$$\gamma(n, j, m) = \gamma(n, j)\left[\frac{c_{jm}\mathcal{N}(x_n, \mu_{jm}, \Sigma_{jm})}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(x_n, \mu_{jm}, \Sigma_{jm})}\right] \tag{3.12}$$

Then, during the M-step the parameters mean (μ_{jm}), covariance (Σ_{jm}) and mixture weights (c_{jm})

of the GMMs of each state j are re-estimated using $\gamma(n, j, m)$ in the following way:

$$\hat{c}_{jm} = \frac{\sum_{n=1}^{N} \gamma(n, j, m)}{\sum_{n=1}^{N} \sum_{m=1}^{M} \gamma(n, j, m)} \tag{3.13}$$

$$\hat{\mu}_{jm} = \frac{\sum_{n=1}^{N} \gamma(n, j, m) x_n}{\sum_{n=1}^{N} \gamma(n, j, m)} \tag{3.14}$$

$$\hat{\Sigma}_{jm} = \frac{\sum_{n=1}^{N} \gamma(n, j, m)(x_n - \hat{\mu}_{jm})(x_n - \hat{\mu}_{jm})^T}{\sum_{n=1}^{N} \gamma(n, j, m)} \tag{3.15}$$

There are two different approaches to estimate parameters of GMMs. In the first approach, the training starts with a single Gaussian (i.e. $M = 1$) per state and the parameters of the Gaussian of each state are estimated by a few EM steps. This is followed by gradual increment of number of Gaussian with parameter re-estimation at each increment step. The second approach starts with required number of Gaussian M e.g., obtained by K-Means (Hartigan, 1975) and the parameters of the GMMs are re-estimated during EM training until convergence. Since the result of the EM training is sensitive to initial parameters (i.e., if the initialization is bad then the resulting models will also be bad), the second approach can be adopted if the segmentation of the training data is available in terms of the HMM states.

In this thesis, we have used HTK toolkit to train some HMM/GMM systems (Young *et al.*, 1997). Given the training data and its transcription (word level or subword unit level), the training procedure starts with a lexicon and list of context-independent subword units. The training is then carried out in the following manner:

1. Initialization of each state of the context-independent subword unit models to a single Gaussian with zero mean and unit variance.

2. EM training with context-independent subword unit models.

3. Generation of context-dependent subword unit models and EM training with context-dependent subword unit models.

4. Tying of the context-dependent subword unit models followed by EM training with tied models.

5. Gradual increment of the number of mixtures and EM training at each increment.

HMM/GMM systems are trained in the maximum likelihood framework i.e., the objective function is the likelihood of the observed data given the parameters. During training, the parameters are chosen so as to increase the likelihood of the data. It is important to note that such estimation criteria does not take into account the competing classes. For practical reasons, it is assumed that the dimensions of the acoustic feature vector are uncorrelated i.e., Σ_{jm} is a diagonal covariance matrix which ignores correlation between acoustic vectors. The correlation is only modelled indirectly through the mixture variable m in a linear fashion. Connectionist or artificial neural network (ANN) methods provide one way to reduce system dependence on such assumptions.

3.5 Hybrid HMM/ANN ASR System

Multilayer perceptrons (MLPs) are the most common ANN architecture that are used for ASR (Bourlard and Morgan, 1994; Morgan and Bourlard, 1995). Some alternative structures are radial basis functions (Renals, 1988), recurrent neural network (Robinson and Fallside, 1991), or time-delay-neural-network (Waibel *et al.*, 1989). The MLPs have a layered architecture with an input layer, zero or more hidden layers and an output layer (Bishop, 1995; Morgan and Bourlard, 1995). Each layer has a certain number of nodes which are connected to the nodes in another layer through weights in a feed forward fashion i.e., the input layer feeds into hidden layer and the hidden layer feeds into output layer. Each layer computes a set of linear discriminant functions followed by a nonlinear activation function (e.g., sigmoid). The nonlinear activation function has two different roles, in the hidden layer it serves to generate higher order moments of the input and in the output layer it acts as a differentiable approximation to the decision threshold of a threshold logic unit (Morgan and Bourlard, 1995). In practice, the nonlinear function in the hidden layer is a sigmoid function while in the output layer is a softmax function.

When the MLP is trained in a classification mode, they have been shown under certain conditions to be capable of estimating the a posteriori probabilities of output class conditioned on the input (Richard and Lippman, 1991; Morgan and Bourlard, 1995). Hence, if the input to the MLP is the acoustic feature vector x_n, the nodes in the output layer correspond to the states $q_n \in \{1, \cdots k, \cdots K\}$ of the HMMs (K nodes in the output layer) and the MLP is "well" trained to discriminate between the states q_n, then the output layer estimates $P(q_n = k|x_n)$, $1 \leq k \leq K$. Thus, applying Bayes' rule

$$\frac{p(x_n|q_n = k)}{p(x_n)} = \frac{P(q_n = k|x_n)}{P(q_n = k)} \tag{3.16}$$

where $P(q_n = k|x_n)$ is the output of the ANN for state k, also referred to as local posteriors and $P(q_n = k)$ is the prior probability of state k. The fraction on the left hand side is called scaled-likelihood. In hybrid HMM/ANN based ASR system emission probability $b_j(x_n)$ in (2.12) is replaced by the scaled-likelihood, since, during recognition the scaling factor $p(x_n)$ is constant for all states and will not affect the classification. To summarize, in hybrid HMM/ANN based ASR systems the emission distribution is modelled by an ANN and the emission probabilities are estimated from the output of the ANN using (3.16).

The main advantages of using an ANN to model emission distribution of HMM are the following:

- Discriminative training at frame level i.e., the ANN is trained to discriminate between the HMM states $q_n \in \{1, \cdots k, \cdots K\}$.
- The input to the ANN can be continuous valued, discrete valued or a mix of both. For instance, the discrete valued gender information (male or female) can be fed into the ANN along with standard continuous feature x_n (Konig *et al.*, 1991).
- The capability of the hidden layer to model higher order moments helps in modelling correlation with-in an acoustic feature vector, and across acoustic feature vectors over time by feeding acoustic feature vectors of more than one frame (as described later).
- Combining multiple streams of information i.e., if there are different streams of information then, for each stream of information different ANN can be trained and, during recognition posterior estimates of all the ANNs can be combined to get a better robust estimate of the emission probability (Hagen, 2001; Misra *et al.*, 2003).

In state-of-the-art hybrid HMM/ANN ASR system, the input to the ANN at any time frame n is typically nine frames of acoustic feature vectors consisting of acoustic feature vector at time frame n, and the preceding and following four frame acoustic vectors i.e., the input is X_{n-4}^{n+4}. The nodes in the output layer of ANN represent K context-independent phoneme units. For English language K is around 42, while languages such as French it is around 36.

Training of an ANN consists of adjusting the weights of the ANN in order to minimize the error between the predicted output vector (output of the ANN) and the desired output vector in a supervised manner. The training of the ANN can be done efficiently by error back propagation algorithm (Rumelhart *et al.*, 1988) that uses a gradient approach to iteratively minimize a cost function. The error functions that are commonly used are mean square error criterion (MSE) and cross entropy error (also called relative entropy error). The use of cross entropy error function is preferred over MSE because cross entropy speeds up the convergence, the correction resulting from this criteria is always linear and does not saturate when the output values are the extremes of nonlinear function and leads to better classification rate (Bishop, 1995).

In order to train an ANN for ASR, we need a desired output vector of dimension K corresponding to each feature vector x_n. The desired vector can be estimated in two different ways:

1. Estimating $\gamma(n, k)$ by forward-backward algorithm (Hennebert *et al.*, 1997).

2. Generating segmentation through Viterbi algorithm (Morgan and Bourlard, 1995).

In other words, similar to HMM/GMM system, the hybrid HMM/ANN based ASR system can be trained by (1) forward-backward training or (2) embedded Viterbi training.

In practice, the embedded Viterbi training is used to train hybrid HMM/ANN systems. In order to speed up the training procedure, the training starts with an initial segmentation of the training data. The initial segmentation can be obtained in different ways:

- Manual segmentation of the training data. This is practical for small training sets (e.g., 1-2 hrs) not for large training sets (e.g., 300 hrs).

- Obtaining segmentation of the training data using an already trained ASR system by Viterbi algorithm (also referred to as forced alignment).

- Linear segmentation of the training data.

The parameters of hybrid HMM/ANN ASR systems are transition probabilities a_{ij}, weights of the trained ANN, and the prior probability of each output unit of the ANN (needed when estimating scaled-likelihood). In forward-backward training, the priors can be estimated by summing $\gamma(n, k)$ over the whole training data for each k and, in embedded Viterbi training the priors can be simply estimated by hand count on the segmentation of the training data.

In this thesis, all ANNs are trained with acoustic vector 9 frames of acoustic vector as input. The output is generally the number of context-independent phonemes for a given task. Given the training data and the segmentation, the training of an ANN starts with initialization of the weights and an initial learning rate:

1. The update of the ANN weights is made after every training example, i.e. online training. The desired vector is one-hot-encoding in which the target class is assigned a probability of 1.0 and all others 0.0.

2. A separate data set which is not part of the training data is used for cross validation in order to avoid over training of the ANN. After each iteration, the performance is evaluated over both cross validation data and training data. If the performance improves on the cross validation data then the training is continued else, the learning rate is reduced by a factor of two for the next iteration.

3. The training continues until the learning rate falls below a certain threshold.

When there is no segmentation available for the training data, the embedded training approach is employed starting with linear segmentation.

3.6 TANDEM System

HMM/GMM-based ASR systems and hybrid HMM/ANN-based ASR systems have been widely studied (Rabiner and Juang, 1993; Bourlard and Morgan, 1994). HMM/GMM models are trained to maximize the likelihood of the data X, where as, an HMM/ANN model is trained to discriminate between the states so as to yield the posterior probability of state q_n.

A TANDEM system combines the discriminative feature of an ANN with Gaussian mixture modelling by using the processed posterior probabilities obtained from the output of ANN (referred to as tandem features) as the input feature for the HMM/GMM based systems. Figure 3.1 illustrates the TANDEM system. This approach has been shown to yield significant improvement over conventional HMM/GMM ASR system using cepstral features in both clean and noisy conditions (Hermansky *et al.*, 2000).

The TANDEM system in spirit is similar to an approach proposed earlier in (Bengio *et al.*, 1992) for speech recognition where, the outputs of ANN was used as observations for HMM/GMM system. This system had three levels, (a) the first level consisted of ANNs trained to recognize broad phonetic classes, (b) the second level consisted of an ANN integrating the outputs of the ANNs of the first level, this ANN was trained to principal components of lower levels, (c) at the third level, the output of the second level ANN was modelled by HMM/GMM system. The Gaussians of GMMs had diagonal covariance matrix. This system yielded better phoneme recognition performance than standard HMM/GMM system and hybrid HMM/ANN system. Furthermore, the phoneme recognition performance improved when the parameters at all the levels were jointly optimized. As we will see later in this section, in TANDEM system the parameters of the ANN and HMM/GMM system are optimized separately and, the ANN output is decorrelated in a different way before being fed into HMM/GMM system.

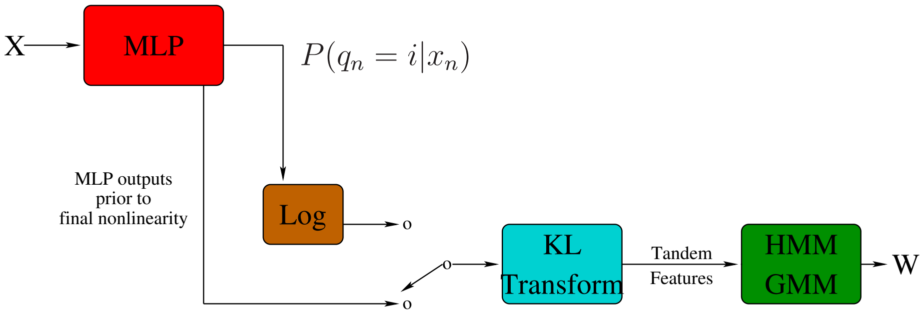


Figure 3.1. Block diagram of TANDEM system.

The TANDEM system is trained in the following manner (Hermansky *et al.*, 2000).

1. An ANN is trained to discriminate between a set of class labels, such as, phonemes. The ANN can be trained with the training data of the intended ASR task (task-dependent training data) or training data of any other ASR task (task-independent training data) (Hermansky *et al.*, 2000). In our studies, the ANN is always trained with task-dependent data.

2. After training the ANN, the task-dependent training data is passed through the ANN to estimate the phoneme posterior probabilities.

3. Since the posterior probabilities obtained from the output of the ANN are skewed, their logs are taken. An alternative is to take the output of the ANN prior to the output layer nonlinearity.

4. Principal component analysis (PCA) is performed on the features obtained in the previous step. The features are then decorrelated by projecting them along the eigenvectors. We refer to the resulting features as tandem-features.

5. HMM/GMM ASR system with diagonal covariance matrices for the Gaussians is then trained with the tandem-features.

During recognition, as for training, the test data is passed through the ANN. The log posterior probabilities obtained are decorrelated by Karhunen-Loeve-transform (KLT) using the PCA statistics collected during training to obtain the tandem-features. The tandem-features are then fed to the trained HMM/GMM ASR system and decoding is performed.

TANDEM systems have several advantages, such as:

- Better use of the different probabilistic basis of the two systems and approaches developed for them.
- It provides a framework where data from different databases could be used together. For instance, if there is not sufficient task-dependent training data to train ANN then a well trained ANN on a different database can be used for tandem-feature extraction (Hermansky *et al.*, 2000; Sivadas and Hermansky, 2004).
- TANDEM systems can be used to combine different features or streams of information efficiently system (Hermansky and Sharma, 1998; Zhu *et al.*, 2004; Ikbal *et al.*, 2004b) similar to hybrid HMM/ANN system. For instance, in (Ikbal *et al.*, 2004b) two ANNs corresponding to features MFCCs and PAC-MFCCs were trained to classify phonemes. The phoneme posterior estimates from the two ANNs were combined through entropy combination approach (Misra *et al.*, 2003) yielding a new estimate of phoneme posterior probabilities. The new estimate of phoneme posterior probabilities were used to extract tandem-features. The resulting tandem-features were used as the input feature to HMM/GMM system. This approach led to improvement in the performance of the ASR system mainly in the noisy conditions.
- The tandem-features exhibit less speaker variability (Zhu *et al.*, 2004). This is due to the ability of the ANN to project the standard acoustic feature on dimensions carrying more speech information. For example, due to speaker variability the standard acoustic feature vectors corresponding to the same phoneme class may be located at different points in the feature space, but may have similar phoneme class probabilities (output of ANN). Thus, we can expect that the acoustic feature vectors of the same phoneme from different speakers to be mapped to same point in the trained ANN's output space.

3.7 Dynamic Bayesian Network Based ASR System

Bayesian networks (BNs) model a set of variables V. The variables can be both discrete and continuous. DBNs extend this framework by modelling these variable at every discrete time step n. DBNs are generalization of HMMs (Zweig, 1998; Stephenson, 2003), and are also part of larger group of probabilistic models called graphical models. From a graphical viewpoint, these variables are the vertices in a directed acyclic graph with edge between the vertices, as illustrated in Figure 3.2.

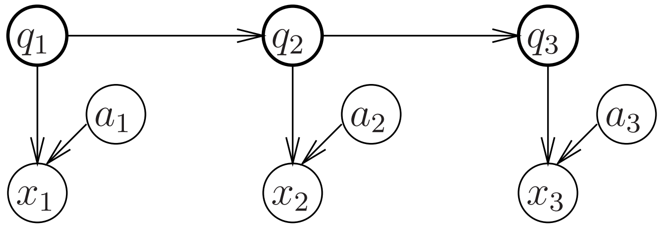


Figure 3.2. Example of DBNs. Note: there is a difference between the visual representation of DBNs and HMMs, for example the vertices of DBNs represent the variable where as the vertices of HMMs are the value of the variable.

The edges have a parent-child relationship, i.e., each edge points from the parent vertex to the child vertex, for e.g., vertex q_1 is parent of vertex x_1. In our work, the edges do not span back in time and they span at most one time frame. Edges from continuous variables go to only continuous variables. If $\mathrm{pa}(v)$ is all the parents of an arbitrary vertex v and $P(v|\mathrm{pa}(v))$ is the local probability distribution associated with vertex v. The joint probability distribution of V is then the product of all the local probability distributions, as shown below:

$$P(V) = \prod_{v_n^i \in V} P(v_n^i|\mathrm{pa}(v_n^i))$$

Thus, for Figure 3.2 we have

$$p(V_1^3) = P(q_1) \cdot P(q_2|q_1) \cdot P(q_3|q_2) \cdot p(a_1) \cdot p(a_2) \cdot p(a_3) \cdot p(x_1|q_1, a_1) \cdot p(x_2|q_2, a_2) \cdot p(x_3|q_3, a_3)$$

where, $V_1^3 = \{q_1, x_1, a_1, q_2, x_2, a_2, q_3, x_3, a_3\}$.

The actual estimation of $p(V_1^3)$ without any statistical assumptions of the dependencies between variables would have needed much more local probability distributions[1] or in other words the DBN representing the actual estimation of $p(V_1^3)$ will have much more edges than in the figure. Thus, one of the main purposes of DBNs is sparse factorization of the joint distribution by learning certain dependencies between the variables. Similar to forward-backward algorithm in HMM, the probabilistic inference consists of a two pass inference: the first to compute the likelihood of the observed data given the prior distribution and the second to compute the posterior distribution of variables given the observed data. This posterior distribution is then used in the EM training as the expected counts. During recognition the data likelihood obtained from the first pass is used to get the most likely sequence of words. In case of the HMMs the probabilistic dependencies and inference are determined at compile time, where as in DBNs this done at run time. This makes DBNs more flexible in the sense that at each time, if we want to change variables or the statistical dependencies, we do not have to write a new program.

DBNs have been recently used in ASR research (Zweig, 1998; Bilmes, 1999; Bilmes and Zweig, 2002; Zweig *et al.*, 2002; Livescu *et al.*, 2003; Stephenson *et al.*, 2004; Bilmes, 2004). In this thesis, we have used the DBN software developed by Todd Stephenson in his PhD thesis (Stephenson, 2003). For further details about the implementation and probabilistic inference process refer to (Stephenson, 2003, Chapter 3, Chapter 5 Section 5.3 and Appendix B). The components of the DBN software that are used in this thesis are dbnExpect, dbnMax (used for training) and dbnVite (used for recognition). The dbnExpect is the E-step of the EM training, which collects the posterior values for each of the hidden discrete variable such as, transition variable and the mixture component variable. The dbnMax is the M-step of the EM training, where the distribution of the each variable distribution jointly with its parents is maximized according to all the posterior counts and then the conditional distribution of each variable given its parent is obtained. Before saving the conditional distributions, the variances of the acoustic feature vector are floored to 0.1 times the global variance of the training data. The dbnVite performs Viterbi decoding in the DBN framework. It uses a simple language model with equal probability to transit from any word to any other word. Furthermore, it does not incorporate word insertion penalty and language scaling factor which are used in standard decoders such as, HDecode in HTK (Young *et al.*, 1997).

In this thesis, the DBN-based ASR systems are trained in the following manner:

1. Initialization: Using the segmentation of the training set (also used to train ANNs), the acoustic feature vectors for each state are clustered into the required number of mixtures for the GMMs. The mean vector and variance vector for each cluster is computed. The variances are then floored so that they are at least 0.1 times the global variance. The GMMs of that state are then initialized with the mean vectors and covariance matrices.

2. One iteration of EM training of the DBNs is performed, i.e. dbnExpect followed by dbnMax.

[1] $p(V_1^3) = P(q_1) \cdot P(q_2|q_1) \cdot P(q_3|q_1, q_2) \cdot p(a_1|q_1, q_2, q_3) \cdot p(a_2|a_1, q_1, q_2, q_3) \cdot p(a_3|a_1, a_2, q_1, q_2, q_3) \cdot p(x_1|a_1, a_2, a_3, q_1, q_2, q_3) \cdot p(x_2|x_1, a_1, a_2, a_3, q_1, q_2, q_3) \cdot p(x_3|x_1, x_2, a_1, a_2, a_3, q_1, q_2, q_3)$

3. After each iteration, the difference between the log likelihoods outputted by dbnMax between two successive iterations is computed. If the difference is above 0.1% then another iteration of EM training is performed else the training ends. This convergence criteria has been chosen so as to have the DBNs that are reasonably trained and at the same time they are trained in a reasonable amount of time.

3.8 Different Tasks and Experimental Setup

In this thesis, we have studied different tasks, namely, isolated word recognition task, spontaneously spoken connected word recognition task and continuous speech recognition task. In this section, we describe the different databases used for these studies and their experimental set up.

3.8.1 Isolated Word Recognition Task

We use the PhoneBook speech corpus for speaker-independent task-independent, small vocabulary (75 words) isolated word recognition (Pitrelli *et al.*, 1995). The definition of training, validation and test set is similar to the one defined in (Dupont *et al.*, 1997). We use the smaller training set of 19421 utterances (243 speakers) and 6598 utterances (96 speakers) for testing. The training set contains 21 list of words and the test set contains 8 lists of words. Each test set list consists of 75 or 76 words. The words and speakers present in the training data do not appear in cross validation and testing data, and vice versa. There are 42 context-independent phones including silence, each modelled by a single emitting state in the systems trained on PhoneBook corpus.

The acoustic vector x_n comprises MFCCs extracted from the speech signal using a window of 25 ms with a shift of 8.3 ms. Cepstral mean subtraction and energy normalization are performed. Ten Mel frequency cepstral coefficients (MFCCs), the first-order derivatives (delta) of the ten MFCCs and the c_0 (energy coefficient) are extracted for each time frame, resulting in a 21 dimensional acoustic vector.

3.8.2 Numbers Task

The OGI-Numbers database contains spontaneously spoken free-format numbers over telephone channel (Cole *et al.*, 1994). The definition of the training set, validation set, and test set is similar to the one defined in (Mirghafori and Morgan, 1998). The training set contains 3233 utterances (approximately 1.5 hours) spoken by different speakers and the validation set consists of 357 utterances. The test set contains 1206 utterances. The vocabulary consists of 31 words with a single pronunciation for each word.

The acoustic vector x_n comprises PLP cepstral coefficients (Hermansky, 1990) extracted from the speech signal using a window of 25 ms with a shift of 12.5 ms, followed by cepstral mean subtraction. At each time frame, 13 PLP cepstral coefficients, their first-order and second-order derivatives are extracted, resulting in 39 dimensional acoustic vector. There are 24 context-independent phonemes including silence.

3.8.3 Continuous Speech Recognition

We have used two different corpus for continuous speech recognition, namely resource management (RM) corpus and conversation telephone speech (CTS) corpus.

Continuous Speech Recognition (DARPA RM) Task

The RM corpus consists of read queries on the status of Naval resources (Price *et al.*, 1988). The task is artificial in many aspects such as speech type, range of vocabulary and grammatical constraint. The training set consists of 3,990 utterances spoken by 109 speakers corresponding to approximately 3.8 hours of speech. Of this, we use 2,880 utterances for training and 1,100 for cross validation and development. The test set contains 1,200 utterances amounting to 1.1 hours in total. The test set is completely covered by a word pair grammar included in the task specification which is used for recognition. There are 44 phonemes including silence. The feature vector x_n comprises PLP cepstral coefficients, their deltas and delta-deltas using a window of 30 ms with a 10 ms frame shift.

Conversational Telephone Speech Task

The training set for conversational telephone speech (CTS) task contains 32 hours of gender balanced CTS speech randomly selected from the Fisher Corpus and the Switchboard Corpus. The tuning/test set was a subset selected from the the NIST 2003 evaluation set. Only those utterances that covered the top most frequent 1000 words with lower than 10% out-of-vocabulary rate were selected, resulting in 2.5 hours of data which was further divided into a 1.2 hour tuning set and a 1.3 hour test set. The tuning and test sets contained similar ratio of the number of utterances from Fisher corpus to the number of utterances from the Switchboard corpus.

3.8.4 Evaluation of ASR Systems

In ASR research, speech recognition system evaluation is performed for two major reasons: (1) to assess the performance of the speech recognition system and (2) if there is more than one speech recognition system, how to compare them?

Generally, the speech recognition systems are evaluated on an unseen test set (data not used during training) in terms of word error rate (WER). The output of the speech recognition system is a sequence of words (automatic transcription). Given the reference sequence of words (ground truth), the evaluation of ASR system is done by comparing the two sequence of strings. This is usually done by computing the Levenshtein distance or the edit distance. The Levenshtein distance[2] between two strings is the minimum number of changes that has to be made in one string to transform it into another (Sankoff and Kruskal, 1999). The changes are basically insertion, deletion, or substitution. The WER is then the Levenshtein distance or edit distance between the ground truth and automatic transcription normalized by the length of ground truth, i.e., if N_r is the number of words in the ground truth and the number of insertion, deletion, and substitution are I, D and S, respectively then the WER is estimated as (in terms of percentage)

$$WER = \frac{I + D + S}{N_r} 100 \tag{3.17}$$

In case of isolated word recognition task, there are no insertion or deletion errors. There are only substitution errors. In this thesis, we evaluate our systems in terms of WER.

Though, the WER is the popular measure to evaluate ASR systems, this measure may not allow clear interpretation of the results in terms of the end usability and can be misleading. If H is the number of words recognized correctly then, $N_r = H + S + D$. Since I is not part of the denominator the WER can be greater than 100%. In other words, there is no upper bound on WER. The insertion penalty in the decoder typically keeps the insertion error low. Similarly if $H = (S+D) = I = \frac{N_r}{2}$ then the WER is 100% in spite of recognizing $\frac{N_r}{2}$ words correctly. More recently other ways to evaluate ASR systems have been proposed such as, word information lost (Morris *et al.*, 2004) and the use of F-measure from information retrieval studies for ASR evaluation (McCowan *et al.*, 2005). In the later evaluation scheme, it has been shown that the word information lost measure is basically product of precision and recall. These evaluation measures truly lie between [0,1] and, make the analysis and the interpretation of the recognition results easy.

[2]In edit distance it is the weighted sum.

When comparing between different speech recognition systems, the question of how much better one system is than another is answered by a statistical significance test (Snedecor and Cochran, 1989, Chapter 5). The statistical significance test starts with an hypothesis that the two systems being compared are equivalent. This is called the null hypothesis. Given a confidence interval, we then determine the trueness of this hypothesis. In the literature, different statistical significance tests have been proposed, such as, proportion test, McNemar test (McNemar, 1947; Snedecor and Cochran, 1989; Gillick and Cox, 1989). The proportion test assumes that the decision taken by each model on each test example are independent and can be modelled by a Binomial distribution, whereas, the McNemar test considers only the test examples on which the two systems disagree (assumes Normal distribution). Both approaches assume that the test sets of two systems are different, but they come from the same distribution. Moreover, they also assume that error of any one of the system is the average of some random variable (the error) estimated on each test example. This average tends to be Normally distributed as the number of test example grows (Keller *et al.*, 2004). In case of ASR system, the later assumption holds true only in the context of isolated speech recognition task, not for continuous speech recognition task. The WER of a continuous speech recognition system is not an average of WER of each test utterance, as can been seen in (3.17). Hence, these approaches can not be used for comparing two continuous speech recognition systems.

In more recent studies, an approach has been proposed to compare systems with error measures such as WER, F_1 measure etc (Bisani and Ney, 2004; Keller *et al.*, 2004). As opposed to the above described approaches (proportion test and McNemar test), this approach assumes that the test set used for testing the systems are the same. The main advantage of this approach is that it does not assume anything about the type of distribution for errors i.e., the empirical distribution is estimated. This is good for error measures such as WER as they may not follow a particular distribution. However, not assuming anything about the type of distribution for errors also puts onus on the availability of the data to yield a robust estimate of the empirical distribution. Given the difference of errors made by the systems on each utterance of the test set and the confidence interval, this approach starts with a null hypothesis that the two systems are statistically equivalent i.e., the mean of the difference of errors is zero. A bootstrap[3] estimate of the distribution of the difference of errors is then obtained. If the null hypothesis lies with in the confidence interval then it is accepted else rejected[4]. All the significance tests in this thesis have been done using this approach.

3.9 Summary

In this chapter, we briefly described the three basic problems in HMM-based ASR, namely, estimation, decoding and training. We described different state-of-the-art ASR systems are being used in this thesis:

(a) HMM/GMM system where, the emission distribution is modelled by GMMs.

(b) Hybrid HMM/ANN system where, the emission distribution is modelled by an ANN

(c) TANDEM system which extracts feature through an ANN trained to classify speech classes (tandem features) and, using the tandem-features as an input to a HMM/GMM system.

(d) DBN-based ASR system which allows flexibility in modifying underlying probability distributions through a single software. This is particularly beneficial when integrating auxiliary knowledge sources where, we need to modify the underlying probability distribution depending upon different assumptions.

[3] Bootstrap is a method to determine the trustworthiness of a statistics. This is done by creating a replica of the statistics by random sampling from the data set with replacement (Efron and Tibshirani, 1993).

[4] The alternate hypothesis that the two systems are statistically different is accepted.

We also present a brief summary of how these systems are trained and tested. We gave an overview of the different tasks that have been studied and, the databases used for these tasks along with experimental setup. Finally, we briefly described how the systems have been evaluated in the present work. In the following chapter, we introduce the notion of auxiliary features and study different ways to intgerate it in standard ASR.

Chapter 4

Auxiliary Features for CI Phoneme-Based ASR

4.1 Introduction

State-of-the-art ASR systems use features $X = \{x_1, \cdots x_n, \cdots x_N\}$ typically derived from the smoothed spectral envelope of the speech signal e.g., linear prediction features, MFCC, PLP features. We refer to these features as "standard" features. These standard features are typically assumed to be conditionally independent identically distributed (c.i.i.d assumption in Page 11). These features are sensitive to different variabilities present in the speech signal such as, speaker variability environmental variability, leading to poor acoustic modelling and degradation in the performance of ASR. The standard acoustic features typically capture short-term (10 ms - 30 ms) information, while ignoring other information/knowledge sources, such as, voice source characteristics, prosody, etc. In this chapter, we study different ways to introduce alternate features, such as,voice source characteristics (pitch frequency), rate-of-speech in order to improve the performance of the ASR (Magimai.-Doss *et al.*, 2003a; Stephenson *et al.*, 2004; Magimai.-Doss *et al.*, 2004b). We refer to these alternate features as "auxiliary features".

Auxiliary features bring additional information, complimentary to usual features or models. This information though can be used as additional feature or conditional variable, i.e., conditioning the emission distribution. The main idea behind integrating auxiliary features in the standard ASR system is to handle the variability present in the speech signal. Thus, yielding better acoustic models. In this work, we examine auxiliary features that are extracted directly from the speech signal according to "non-conventional" ways e.g., focussing on long-term properties, prosody.

This chapter is organized as follows. We explain the notion of the auxiliary feature in more detail and summarize the past work made in this direction in Sections 4.2 and 4.3. Section 4.4 presents the different ways to model the auxiliary features in state-of-the-art ASR systems and, Section 4.5 gives the implementation details. In Section 4.6, we present the different auxiliary features that have been examined in this thesis. The Section 4.7 briefly summarizes the studies conducted in the framework of HMM/DBN-GMM system. Section 4.8 presents the experimental studies conducted on different databases in the framework of hybrid HMM/ANN system using context-independent (CI) phonemes as subword units. We finally conclude in Section 4.10.

4.2 Auxiliary Feature

As seen in Section 2.3 and Chapter 3, HMMs are mainly based on the estimation of local likelihoods, i.e.:

$$p(x_n|q_n) \tag{4.1}$$

As we saw in the Chapter 3, this likelihood can be estimated using GMMs or ANNs. When using standard cepstral features x_n we hope that the distribution of the hidden states q_n are well separated and thus, allows good discrimination. However, the standard cepstral features are sensitive to different variabilities present in the speech signal (described earlier in Chapter 1) thus, leading to large variabilities that must be accounted by the acoustic model.

By modelling relevant auxiliary feature a_n, we can improve the robustness of acoustic model to variabilities present in the speech signal, using enhanced likelihood:

$$p(x_n, a_n|q_n) \tag{4.2}$$

Depending upon the relevance of a_n and the reliability of its measurement or estimation (during training and/or recognition), its direct use in estimation of (4.2) (during training and/or recognition) may improve or degrade the performance of the resulting system. For instance, one of the most common auxiliary feature that is used in state-of-the-art ASR is gender information (Konig *et al.*, 1991; Vergin *et al.*, 1996). In the case of gender modelling the auxiliary feature a_n is a binary variable, which is directly used during training (where we assume that we know the gender of speaker). However, during recognition one can explicitly estimate the speaker's gender and select the acoustic models or, infer the gender automatically as a by-product of recognition process by picking the conditional model that yields the highest likelihood.

We can integrate the auxiliary feature a_n in standard HMM-based ASR in the following ways:

(a) Augmenting the standard features x_n with auxiliary feature a_n and, estimating the emission distribution as:

$$p(x_n, a_n|q_n) = p(x_n|a_n, q_n) \cdot p(a_n|q_n) \tag{4.3}$$

(b) Conditioning the emission distribution upon a_n:

$$p(x_n, a_n|q_n) \approx p(x_n|a_n, q_n) \cdot p(a_n) \tag{4.4}$$

Assuming equal prior probabilities $p(a_n)$, a particular, well know and successful, application instance of 4.4 is gender modeling where, $a_n \in \{male, female\}$. In this case, the usual implementation of the conditional density is to simply use two separate HMM models for male and female (Konig *et al.*, 1991; Vergin *et al.*, 1996).

While implementing (4.3) is relatively easy, the implementation of a system based upon (4.4) is not so straightforward. In the case of gender modelling, the auxiliary feature is a discrete-valued variable with two values. Thus, we need to only train two acoustic models corresponding to the two values. However, if the auxiliary feature is multi-valued or continuous-valued then it is harder to implement a system based upon (4.4) as, an acoustic model corresponding to each value of the auxiliary feature has to be trained. In case of multi-valued auxiliary feature there are finite number of values for the auxiliary feature and it is feasible to train acoustic models for each value by splitting the training data. However, the splitting of the training data can result in poor acoustic models because of limited amount of training data. Similarly it is easy to see that if the auxiliary feature is continuous-valued then there are infinite values and hence, the implementation is not straightforward. One possible approach that has been suggested in the literature when using GMMs to model emission distribution is: a linear regression over the mean of the Gaussians. We describe this approach in detail later in Section 4.5.

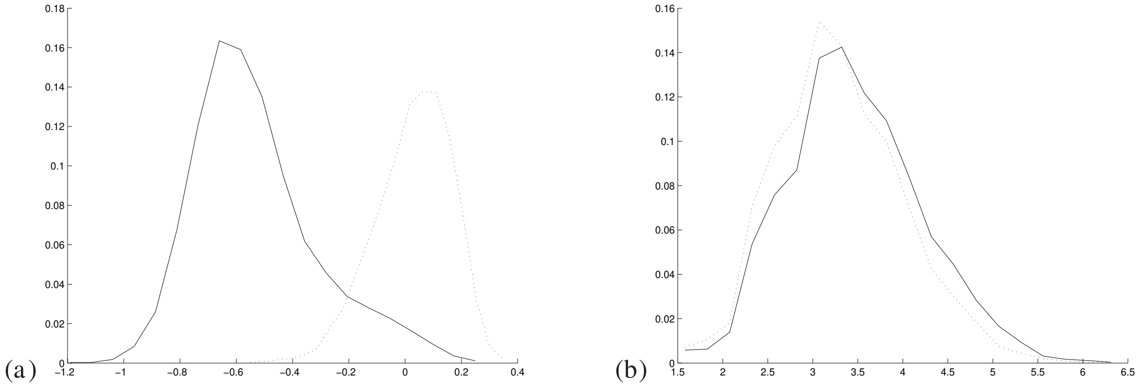


Figure 4.1. Illustration of the *ideal* type of distributions according to our definition of auxiliary information: (a) x_n having different, discriminant distributions (shown are the distributions of the first PLP coefficient for the phonemes /ao/ and /f/); (b) a_n of ROS having similar, non-discriminant distributions (shown are the distributions of ROS for the phonemes /ao/ and /f/). These are normalized, empirical distributions using all of the training data for the respective phonemes, with the segmentation used for EM initialization.

The auxiliary features, when compared to standard features tend to carry higher level information. Standard features carry information that is more relevant to the hidden states, but this may not be case with auxiliary features, as illustrated in Figure 4.1. So, while the standard features are changing as the states change, the auxiliary features may be slowly changing compared to the standard features. The changes in auxiliary features may depict higher level of discourse e.g., prosody.

There are different possible sources of auxiliary features. It can be an information precisely known such as, gender information or the age of the speaker. It can be precise measurements such as, articulator positions. It can be an estimate from the speech signal such as, pitch frequency, short-term energy, rate-of-speech. It can be an estimate from other modalities (e.g., visual) such as, lip contours, mouth opening. Furthermore, the auxiliary feature can be static or dynamic in nature. For example, gender information is a static feature in a single speaker environment, while pitch frequency or short-term energy is a dynamic feature.

4.3 Relation to previous work

The auxiliary feature can be continuous valued or discrete valued. In the past much focus has been towards auxiliary features that do not take many values. Work such as, (Konig *et al.*, 1991) uses an ANN to determine the gender of the speaker. The input to the ANN are cepstral features and the output are the posterior probabilities of the gender given the data. The output of the ANN are treated in two different ways:

- They are used as additional features.
- They are are used to select the gender-dependent acoustic model or to weight the output of the two acoustic models (hiding the gender information) for decoding.

The later approach yielded significant improvement in the performance of ASR.

In (Vergin *et al.*, 1996), the location of the first two formant frequencies was used for gender classification. Using this gender classifier, the training data was split into male and female part and, separate acoustic models were trained. During recognition, the gender classifier was used to select the acoustic model for decoding. This lead to a relative improvement of 14% over a single acoustic model system.

In (Siegler, 1995), the measure of speaking rate was used to adapt the acoustic models, HMM state transition probabilities, language model weight, the dictionary and phoneme set to compensate for the effect of fast speech. Adapting the language weight, HMM state transition probabilities, and dictionary led to an improvement in the performance of ASR.

In (Martinez *et al.*, 1998), two speaking rate dependent HMMs were trained, one corresponding to slow speech and the other corresponding to fast speech. These two models, along with a speaking rate classifier, yielded a 32% reduction of average error rate. While (Martinez *et al.*, 1998) used a discrete valued estimate of speaking rate, (Morgan *et al.*, 1997) used a continuous estimate of speaking rate to divide the test set into four bins corresponding to different speaking rates and, optimized the exit state transition probabilities for each of the bins. This lead to a 14% reduction in word error rate.

In all the above described works, the auxiliary feature was in some way discretized. (Singer and Sagayama, 1992) studied the correlation between a continuous valued auxiliary feature (pitch frequency) and cepstral features. To exploit the correlation between pitch frequency and spectral parameters, they proposed a pitch-spectrum normalization approach where the cepstral vector was normalized by a linear regression over the auxiliary feature. This approach yielded models with lower variance and, improved the separability between the phoneme classes in the acoustic space, as a result of which improved phoneme recognition performance and ASR performance were achieved. More recently, this approach has been extended to condition the acoustic model (as in (4.4)) instead of as a feature level normalization (Fujinaga *et al.*, 2001). Here, the auxiliary feature was used in a regression to better model the Gaussian distribution of the regular features i.e, conditioning the standard features by shifting their first order moment. In their study, they showed the advantage of using auxiliary features as conditioning variable as opposed to the conventional approach where, the auxiliary feature is appended to the standard feature (as in (4.3)). If the auxiliary features are assumed to be correlated with standard features then conditioning the standard features on the auxiliary feature results in Gaussians with reduced variance during training. The auxiliary features investigated were pitch frequency and energy. This approach to condition the emission distribution upon auxiliary feature led to a improvement in phoneme recognition and isolated word recognition tasks. In addition to this, they observed that the approach of appending the auxiliary feature to the standard acoustic feature leads to degradation in the ASR performance..

In (Zweig, 1998), the notion of the auxiliary variable was introduced within the framework of DBNs, where it was referred to as “context” variable. The idea behind using an auxiliary variable was to model contextual information i.e., features in relation to features at the previous time frame and also to model the correlation between the features at the present time frame. The auxiliary variable was a latent variable i.e., hidden (expect in certain experiments where it was initialized to reflect voicing) during both training and recognition. (Zweig, 1998) gave theoretical justifications (but no experiments) behind using auxiliary variable with real data.

In (Stephenson, 2003), this notion of auxiliary variable was furthered using real data within the framework of DBNs. The different auxiliary features studied were articulatory features, pitch frequency, short-term energy and rate-of-speech. Stephenson investigated different ways to introduce auxiliary features in state-of-the-art ASR systems, such as by appending them or by using them to condition the standard features. His work revealed the need for a time-dependent auxiliary feature that conditions the standard features i.e., the auxiliary feature shifts the Gaussians that model the standard features in order to estimate better acoustic models that are robust to noisy conditions. In (Fujinaga *et al.*, 2001) and previous related work described earlier in this section, the auxiliary feature was observed during both training and recognition. Stephenson investigated in detail the idea of observing the auxiliary feature during training and hiding it (i.e. integrating over all possible values) during recognition. It was found that hiding the auxiliary features during recognition sometimes make the acoustic models more robust, especially in noisy conditions.

The initial part of the studies reported in this chapter were carried out with Todd Stephenson. While (Stephenson, 2003), focusses more on the use of DBNs for modelling auxiliary feature, the present work focusses on modelling auxiliary feature in the framework of the hybrid HMM/ANN systems and extending them to state-of-the-art TANDEM systems.

4.4 Modelling Auxiliary Features in Acoustic Models

As described in Section 2.3, standard HMM-based ASR systems model the evolution of the observed acoustic feature sequence $X = \{x_1, \cdots, x_n, \cdots, x_N\}$ and the associated hidden state sequence $Q = \{q_1, \cdots, q_n, \cdots, q_N\}$ through the joint likelihood[1]:

$$p(Q, X) \approx \prod_{n=1}^{N} p(x_n|q_n) \cdot P(q_n|q_{n-1}) \tag{4.5}$$

where $p(x_n|q_n)$ is the local likelihood and $P(q_n|q_{n-1})$ is the state transition probability.

Assuming that we have access to additional auxiliary features a_n associated with x_n, thus yielding auxiliary feature sequence $A = \{a_1, \cdots, a_n, \cdots, a_N\}$, we can integrate the auxiliary feature in the standard HMM-based ASR by modelling the joint likelihood $p(Q, X, A)$.

$$\begin{aligned} p(Q, X, A) &\approx \prod_{n=1}^{N} p(x_n, a_n|q_n) \cdot P(q_n|q_{n-1}) &(4.6)\\ &\approx \prod_{n=1}^{N} p(x_n|q_n, a_n) \cdot p(a_n|q_n) \cdot P(q_n|q_{n-1}) &(4.7)\end{aligned}$$

In the case of hybrid HMM/ANN systems, it is easy to realize a system according to (4.6) by augmenting the feature vector x_n with a_n and, then modelling the evolution of the augmented feature vector over the associated hidden state space Q, similar to (4.5). In other words, the input to the ANN is the augmented feature vector (x_n, a_n). As, ANN can model the correlation between the input features, we can expect that the ANN will model the possible correlation between standard acoustic feature and auxiliary feature. However, in the case of a HMM/GMM system, the system will have to model the dependency between state q_n and auxiliary feature a_n in addition to the emission distribution conditioned by a_n, as can be observed from (4.7).

The above described approach to integrate auxiliary feature implicitly models the dependency between the state q_n and the auxiliary feature a_n, see (4.7). This dependency can be noisy, as illustrated in Figure 4.1 with rate-of-speech, which obviously cannot discriminate between the states at frame level. In such a case, it would be better to relax the joint distribution in (4.7) by assuming independence between a_n and q_n, yielding:

$$p(Q, X, A) \approx \prod_{n=1}^{N} p(x_n|q_n, a_n) \cdot p(a_n) \cdot P(q_n|q_{n-1}) \tag{4.8}$$

The auxiliary feature can also be integrated by reducing (4.7) assuming conditional independence between x_n and a_n given q_n, yielding

$$p(Q, X, A) \approx \prod_{n=1}^{N} p(x_n|q_n) \cdot p(a_n|q_n) \cdot P(q_n|q_{n-1}) \tag{4.9}$$

Formulation (4.9) is similar to appending the auxiliary feature in a HMM/GMM system where each dimension is assumed to be conditionally independent of the others given the state.

It may happen that the auxiliary feature may not always be available. For instance, articulatory measurements which would be available during training, but not during recognition. Also, the estimation of the auxiliary feature may be unreliable. For instance, it has been shown in the literature that pitch frequency estimation is error prone (Bagshaw *et al.*, 1993). The unavailability of the auxiliary feature or unreliable estimate of the auxiliary feature will yield noisy estimates

[1] After a first-order Markov assumption for the state sequence and an c.i.i.d assumption for the feature sequence

of $p(x_n, a_n|q_n)$, which can degrade the performance of the ASR system. In such cases, it may be better to observe the auxiliary features during training and, during recognition hide (marginalize out) them or infer (estimate) them automatically. For instance, in gender modelling during training two acoustic models are trained (one for $a_n = male$ and one for $a_n = female$). During training, the gender information is available. However, during recognition the gender information may not be always available. Then, the decoding can be done in three possible ways:

1. Estimating explicitly the value of a_n by using a gender classifier and, select the acoustic model for decoding.

2. Most often it is not possible to reliably estimate the value of a_n during recognition. In such a case, it is better to integrate all possible values of a_n (hide) as described later in (4.11) and, using the marginalized distribution for decoding.

3. The third possibility is to decode with both the acoustic models separately and, pick the decoded hypothesis that has the maximum likelihood.

The auxiliary feature can be hidden or marginalized out by integration for continuous valued auxiliary features and by sum for discrete valued auxiliary features:

$$p(x_n|q_n) = \int_{-\infty}^{\infty} p(x_n, a_n|q_n)\, \mathrm{d}a_n \quad \text{for } a_n \text{ continuous} \tag{4.10}$$

$$p(x_n|q_n) = \sum_{l=1}^{L} p(x_n, a_n = l|q_n) \quad \text{for } a_n \text{ discrete} \tag{4.11}$$

In our studies, the auxiliary features are always observed during training. During recognition, we have the choice of observing the auxiliary feature or hiding it as described above. When the auxiliary feature is hidden, the resulting estimate of $p(x_n|q_n)$ is used in (4.5) to perform decoding. In the following section, we describe the implementation details of integrating auxiliary feature in different ASR systems based upon (4.7), (4.8), and (4.9).

4.5 Implementation

In this section, the possible ways to integrate auxiliary features in standard (1) HMM/GMM systems (2) hybrid HMM/ANN systems, and (3) TANDEM systems are discussed. We have studied the HMM/GMM system integrating auxiliary features using DBNs (HMM/DBN-GMM). We have used DBNs because DBNs provide a general framework for handling any of the different assumptions described in the previous section to integrate auxiliary features in the same software (Zweig, 1998; Stephenson, 2003). However, in standard HMM/GMM ASR system, different softwares would need to be developed for handling each change in the assumptions for handling a_n.

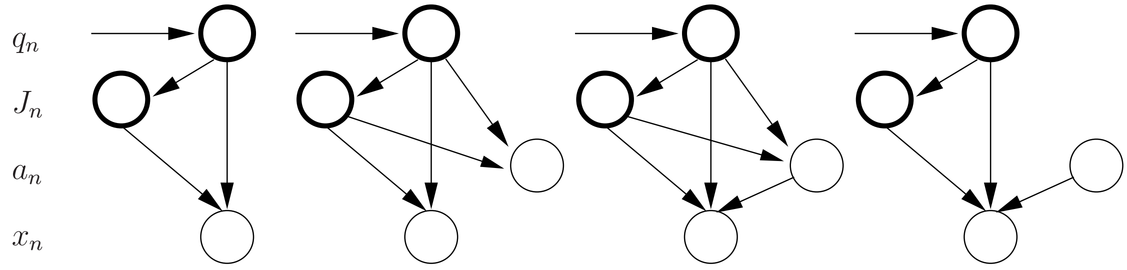
4.5.1 HMM/DBN-GMM Based ASR

Based on the different assumptions described in Section 4.4 to integrate auxiliary features, we consider DBNs with different topologies:

1. HMM/DBN-GMM Baseline system, "Baseline":

   The baseline system with no auxiliary features is based on (4.5). Figure 4.2 (a) shows the BN corresponding to this system. The emission distribution $p(x_n|q_n)$ using GMMs with J mixture components:

$$p(x_n|q_n) = \sum_{j=1}^{J} P(J_n = j|q_n)\, p(x_n|q_n, J_n = j). \tag{4.12}$$



(a) Baseline (b) $x_n \perp\!\!\!\perp a_n \mid \{q_n, J_n\}$ (c) x_n, a_n (no assumption) (d) $a_n \perp\!\!\!\perp q_n \mid x_n$

Figure 4.2. BNs for ASR: (a) has only x_n and does not use auxiliary information; (b) models a_n in the same manner as a standard feature; (c) models a_n as "mid-level" auxiliary information; (d) models a_n as "high-level" auxiliary information. The thick circles represent the discrete variables and the thin represent continuous variable. Figure 3.2 in Page 29 shows the DBN without the mixture variable for the case of (d).

The elements of x_n are standard acoustic features, furthermore, assumed to be statistically independent of each other, given the state and the mixture component, meaning that the mixture components have zeros off of the diagonal of each covariance matrix. Doing so reduces the complexity in the models and allows more robust models to be learned without the need for large amounts of data that very complex models would demand for effective learning. The correlation between the acoustic features is indirectly modelled via the mixture variable j

2. Auxiliary feature is treated as an addition feature, "$x_n \perp\!\!\!\perp a_n \mid \{q_n, J_n\}$"[2]:

   This system integrates auxiliary features based on (4.9), where an independence between x_n and a_n is assumed given q_n and J_n. Figure 4.2 (b) shows the BN of this system. This system is equivalent to appending a_n to x_n and using the single feature vector in a standard HMM. The emission distribution $p(x_n, a_n|q_n)$ of this system with GMMs based on (4.9) is the following:

$$\begin{aligned} p(x_n, a_n|q_n) &= p(x_n|q_n)\, p(a_n|q_n) && (4.13) \\ &= \sum_{j=1}^{J} P(J_n = j|q_n)\, p(x_n|q_n, J_n = j)\, p(a_n|q_n, J_n = j). && (4.14) \end{aligned}$$

   As in the baseline system, all covariance matrices have zeros off of the diagonal.

3. Auxiliary feature is treated as a mid-level feature, "x_n, a_n (no assumptions)":

   Figure 4.2 (c) shows the BN corresponding to the system "x_n, a_n (no assumptions)". The "x_n, a_n (no assumption)" system enhances the "$x_n \perp\!\!\!\perp a_n \mid \{q_n, J_n\}$" by conditioning the elements of the mixture models for a_n upon the respective elements of the mixture models for x_n. This gives conditional Gaussians (Lauritzen and Jensen, 2001) in the mixture components for x_n and regular Gaussians in the mixture components for a_n, letting $p(x_n, a_n|q_n)$ be modeled according to (4.7) as:

$$\begin{aligned} p(x_n, a_n|q_n) &= p(x_n|a_n, q_n)\, p(a_n|q_n) && (4.15) \\ &= \sum_{j=1}^{J} P(J_n = j|q_n)\, p(x_n|a_n, q_n, J_n = j)\, p(a_n|q_n, J_n = j). && (4.16) \end{aligned}$$

   As a_n is continuous valued, $p(x_n|a_n, q_n, J_n = j)$ is modelled by conditional Gaussian. In a conditional Gaussian, the first order moment (mean) of the Gaussian distribution modelling

[2] $x_n \perp\!\!\!\perp a_n \mid \{q_n, J_n\}$ has to be read as x_n independent of a_n given q_n and J_n.

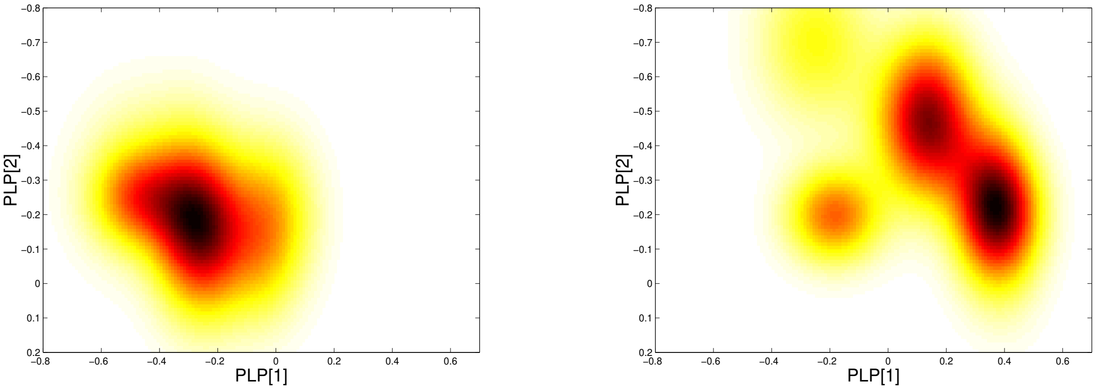


Figure 4.3. Conditional Gaussian mixture models, illustrated by the first state of the phoneme /w/ and the first and second PLP coefficients with energy as the auxiliary variable. These two graphs are taken from a single conditional GMM that was learned from the OGI Numbers data for the "x_n, a_n" system in Table IV of Stephenson *et al.* (2004). On the left is the result of conditioning (instantiating) the (conditional) GMM on a low energy value; on the right is the result of conditioning the same (conditional) GMM on a high energy value. The resulting GMMs after conditioning change with different conditioning values. Furthermore, with different energy values, the covariance (as indicated by the shape of the GMMs) changes as a function of the energy value. Note that the conditional GMM can be viewed as a different (regular) GMM for each instantiation of the conditioning variable.

x_n is shifted according to the value of a_n as shown in below:

$$p(x_n|a_n, q_n = k, J_n = j) = \mathcal{N}(x_n, \mu_{kj} + b \cdot a_n, \Sigma_{kj}) \tag{4.17}$$

where b is the regression coefficient which is estimated during training along with μ_{kj} and Σ_{kj}, e.g. see (Lauritzen and Jensen, 2001; Fujinaga *et al.*, 2001). A conditional GMM can be viewed as a different (regular) GMM for each instantiation of a_n. As descried earlier Σ_{kj} is a diagonal covariance matrix and the covariance between the elements is implicitly modelled through discrete mixture variable. However, by having a_n condition the distribution for x_n, the a_n behaves like a continuous mixture variable i.e., there are infinite number of mixture components with the mixtures closer to the mean of a_n having large weights. Doing so we further implicitly model the covariance between the elements i.e., both continuous valued a_n and discrete valued J_n allow more modelling of x_n's covariance for each state[3] - see Figure 4.3. Furthermore, there will be smaller variance for distribution of x_n, as part of the variance is accounted for by a_n (Fujinaga *et al.*, 2001). If a_n was discrete valued then a GMM corresponding to each discrete value of a_n is trained (e.g., gender modelling).

Note that the use of (4.16) instead of (4.14) represents a small increase in the computation for each mixture. That is, as (4.16) uses conditional GMMs, there is an additional multiplication and addition to shift each mean of x_n according to the value of a_n (assuming a_n's value is available).

4. Auxiliary feature conditions the emission distribution, "$a_n \perp\!\!\!\perp q_n \mid x_n$":

   Equation (4.16) above includes a_n in the state-dependent mixture model. However, our standard way of integrating a_n involves treating a_n independent of q_n as in (4.8). Figure 4.2 (d) shows the BN corresponding to this system. The distribution $p(x_n, a_n|q_n)$ is modeled as:

$$\begin{aligned} p(x_n, a_n|q_n) &= p(x_n|a_n, q_n)\, p(a_n) && (4.18)\\ &= \left[\sum_{j=1}^{J} P(J_n = j|q_n)\, p(x_n|a_n, q_n, J_n = j)\right] p(a_n). && (4.19)\end{aligned}$$

[3] In spirit this approach is similar to other approaches which try to "explicitly" model the covariance between the elements, such as, semi tied covariances (Gales, 1999).

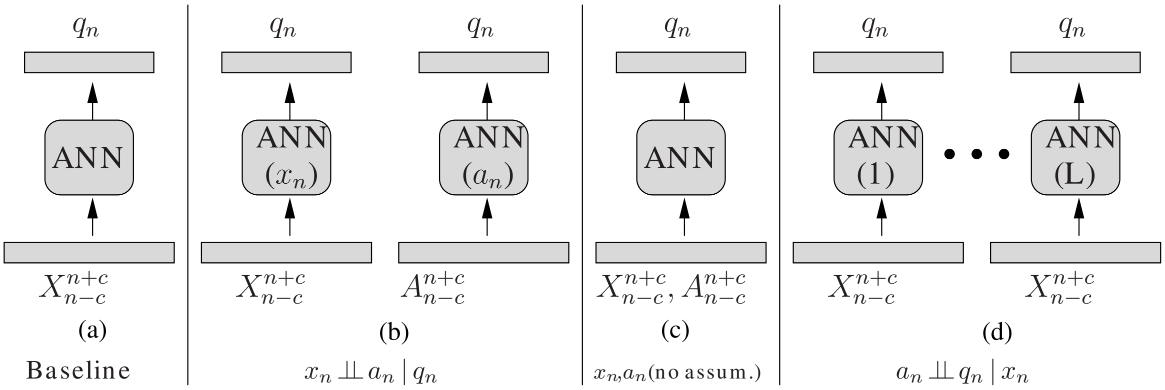


Figure 4.4. ANNs for hybrid HMM/ANN ASR. Presented are (a), the baseline x_n only ANN; (b), the two ANNs treating x_n and a_n as two separate streams; (c), the ANN with a_n appended to x_n and used as a standard feature; and (d), the multiple ANNs using discretized auxiliary information (with L values).

x_n's distribution is still modeled using conditional Gaussian components, as in the "x_n, a_n" system above. However, a_n is given a simpler distribution, i.e., a single Gaussian, outside of the mixture model.

4.5.2 Hybrid HMM/ANN based ASR

As hybrid HMM/ANN ASR is a competitive method compared to HMM/GMM ASR, we also investigate how to integrate a_n in the framework of hybrid HMM/ANN ASR system with similar assumptions to those used in HMM/DBN-GMM system.

1. Hybrid HMM/ANN baseline system, "Baseline":

   Our baseline hybrid HMM/ANN system estimates the scaled-likelihood from (3.16) using the observations for X_{n-c}^{n+c}, with a window size of nine frames (i.e., $c = 4$). Figure 4.4 (a) shows the hybrid HMM/ANN baseline system.

2. Auxiliary feature is treated as a separate stream, "$x_n \perp\!\!\!\perp a_n \mid q_n$":

   In this system, a_n is integrated in the hybrid HMM/ANN based ASR according to (4.9). In order to have q_n condition a_n's distribution but not x_n's, the continuous valued a_n needs to have its layers separated from x_n's layers. We take the approach of having separate ANNs for x_n and a_n, thus being a multi-stream approach (Dupont, 2000). Figure 4.4 (b) shows this system. This most closely resembles the BN in Figure 4.2 (b) yet has a lot more independence between x_n and a_n.

3. Auxiliary feature is treated as an additional feature, "x_n, a_n (no assumptions)":

   In integrating a_n in the HMM/ANN context with no additional assumptions to it, we treat a_n as an additional feature by giving the ANN inputs that have a_n appended to x_n. This is depicted in Figure 4.4 (c), where the input to the ANN is augmented feature vector (x_n, a_n). By inputting these augmented inputs into the same hidden layer, we jointly model the correlation between a_n and x_n and between a_n and q_n (similar to what is done in the BN presented in Figure 4.2 (c)). Equation (3.16) is expanded with the observations for both X_{n-c}^{n+c} and A_{n-c}^{n+c} as shown in (4.20).

$$\frac{p(X_{n-c}^{n+c}, A_{n-c}^{n+c}|q_n)}{p(X_{n-c}^{n+c}, A_{n-c}^{n+c})} = \frac{P(q_n|X_{n-c}^{n+c}, A_{n-c}^{n+c})}{P(q_n)} \tag{4.20}$$

The conditional Gaussians in the DBN framework used for modelling system "x_n, a_n (no assumptions)" model the relation between a_n and x_n in a linear manner. However, the ANNs used for modelling x_n, a_n with no assumptions here in the HMM/ANN framework model the relation between a_n and x_n in a non-linear manner (Bourlard and Morgan, 1994). Therefore, the relation between a_n and x_n can potentially be modelled better with an ANN.

4. Auxiliary feature conditions the emission distribution, "$a_n \perp\!\!\!\perp q_n \,|\, x_n$":

   In treating a_n as an additional input feature above, the hidden layer carried information about a_n to the output layer representing q_n. However, system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" based upon (4.8) assumes independence between q_n and a_n. We achieve this by having a separate ANN for each value of a discretized a_n[4] as shown in Figure 4.4 (d). Each of these separate ANNs has a window size of nine frames for x_n. Thus, the modeling of x_n is done differently depending on which discrete value $1, \ldots, L$ that a_n has; however, the value of a_n does not directly affect q_n (only indirectly via x_n). Likewise, in a HMM/DBN-GMM system, if a_n had been discretized, the effect would have been to have a different set of GMMs for each discrete value of a_n. In preliminary studies with DBNs, (Stephenson *et al.*, 2001) used a discretized a_n with discretized x_n. Here, in hybrid HMM/ANN system, x_n is a continuous variable and a_n is a discrete variable.

4.5.3 TANDEM

Standard ASR systems use features derived from the smoothed spectral envelope of the speech signal as the observation x_n. More recently, TANDEM systems have been proposed which have been shown to perform better than state-of-the-art HMM/GMM ASR system. In TANDEM systems, the cepstral features are transformed into estimates of posterior probabilities using an ANN (Hermansky *et al.*, 2000). These posterior probabilities are then processed and, used as the input feature (tandem feature) for a standard HMM/GMM ASR system. In Section 3.6, we have described the TANDEM system is in detail.

In this section, we propose two approaches to integrate auxiliary features in TANDEM system (Magimai.-Doss *et al.*, 2004b):

- Tandem(CEP+AUX): In this approach, the tandem features are extracted from the ANN of hybrid HMM/ANN systems jointly modelling the cepstral features and the auxiliary features ("x_n, a_n (no assumption)" and "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" described in Section 4.5.2). The HMM/GMM systems are then trained in the conventional way using these tandem features. Figure 4.5 illustrates this approach.

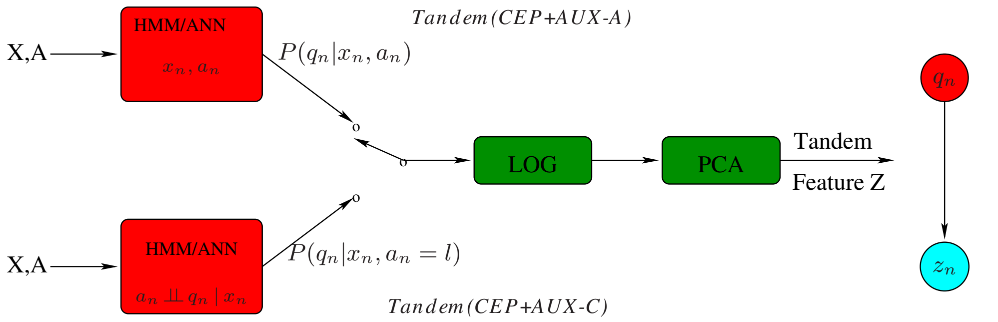


Figure 4.5. Block diagram of the approach Tandem(CEP+AUX) to integrate auxiliary features in TANDEM systems. x_n, a_n hybrid HMM/ANN system corresponds to Figure 4.4 (c) and $a_n \perp\!\!\!\perp q_n \,|\, x_n$ HMM/ANN system corresponds to the Figure 4.4 (d). Z is the tandem feature sequence $\{z_1, \cdots z_n, \cdots z_N\}$. $P(q_n|x_n, a_n)$ is the output of the ANN where the auxiliary a_n is appended to the standard feature x_n. $P(q_n|x_n, a_n = l)$ is the output of the ANN where a_n conditions the emission distribution. As described in the previous section, a_n is quantized to L values and $l \in L$.

[4] If a_n is continuous valued then we need to train infinite number of ANNs corresponding to every instantiation of a_n.

- Tandem(CEP)+AUX: In this approach, the tandem-features are extracted from the ANN of the hybrid HMM/ANN baseline system ("Baseline") and, are then modelled jointly with the auxiliary features in the framework of HMM/DBN-GMM (as described earlier in Section 4.5.1). This approach is illustrated in Figure 4.6.

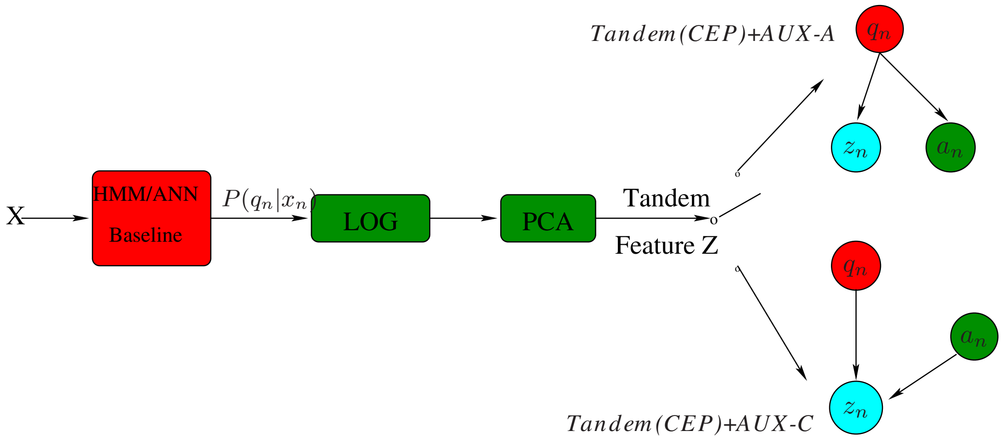


Figure 4.6. Block diagram of the approach Tandem(CEP)+AUX to integrate auxiliary features in TANDEM. "Baseline" hybrid HMM/ANN system corresponds to the system in Figure 4.4 (a). Tandem(CEP)+AUX-A approach corresponds to the HMM/DBN-GMM system for the BN in Figure 4.2 (b) and Tandem(CEP)+AUX-C corresponds to HMM/DBN-GMM system for the BN in Figure 4.2 (d).

4.5.4 Discussion

In the Sections 4.5.1 and 4.5.2, we described how the auxiliary feature a_n can be integrated with different assumptions in HMM/DBN-GMM ASR systems and hybrid HMM/ANN ASR systems, respectively. There are certain differences in the way the two ASR systems handle a_n:

- In the case of system "x_n, a_n (no assumptions)" based on HMM/DBN-GMM system, the auxiliary feature a_n is treated like a mid level information i.e., it is used as additional feature and, to condition the acoustic model. However, in case of system "x_n, a_n (no assumptions)" based on hybrid HMM/ANN system the auxiliary feature a_n is treated like a standard acoustic feature (low level information) i.e., the input to the ANN is the augmented feature vector (x_n, a_n).

- In the case of system "$x_n \perp\!\!\!\perp a_n \,|\, \{q_n, J_n\}$" based on HMM/DBN-GMM system, the auxiliary feature is treated like a standard feature i.e., the input to the system is augmented feature vector (x_n, a_n). However, in case of system "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" based on hybrid HMM/ANN system the auxiliary feature is treated like a separate stream of information. Since our main focus is upon integrating the auxiliary feature as an additional feature or feature conditioning the emission distribution, the multi stream approach will not be the focus of our work.

- In case of HMM/DBN-GMM systems, the auxiliary feature a_n can be hidden in all the systems (Stephenson, 2003, Chapter 4). When using mixture distributions, the auxiliary feature can be hidden by rewriting (4.10) as:

$$p(x_n|q_n) = \sum_{j=1}^{J} \int_{-\infty}^{\infty} P(J_n = j|q_n) \cdot p(x_n, a_n|q_n, J_n = j) \, \mathrm{d}a_n \tag{4.21}$$

  In case of hybrid HMM/ANN systems, except for system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$", it is not obvious how to hide the continuous-valued auxiliary feature a_n. In system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$", the auxiliary feature can be hidden according to (4.11). In case of other systems "x_n, a_n (no assumptions)" and "$x_n \perp\!\!\!\perp a_n \,|\, q_n$", the auxiliary feature a_n can be hidden by treating it as a latent variable similar to (Bridle and Cox, 1991). However, in this work the auxiliary feature a_n in systems "x_n, a_n (no assumptions)" and "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" is always observed.

4.6 Auxiliary Features Examined

In the previous section, we described how auxiliary feature,a_n, can be integrated in a state-of-the-art HMM-based ASR system. In this section, we describe the different auxiliary features that were investigated in this thesis. We are looking at three types of auxiliary feature automatically extracted from the speech signal: (1) (2) pitch frequency (i.e., the fundamental frequency F_0), (2) rate-of-speech (ROS), and (3) short-term energy (in the logarithm domain). They are all fundamental features of speech which change within a given speaker or/and utterance according to prosodic conditions and the environment.

4.6.1 Pitch Frequency

Pitch is a perceptual quantity, but its acoustic correlate (rate of vibration of vocal cords), referred to as fundamental frequency or pitch frequency, can be estimated from the speech signal. The absence or presence of pitch frequency is highly correlated with the phonemes. Hence, there is some relation between the states and this auxiliary feature. However, this relation is between groups of states (voiced vs. unvoiced) and the auxiliary feature. Also, the influence of pitch frequency on formant frequencies of vowels has been observed in perceptual experiments and analytical studies (Hermansky *et al.*, 1983). It has been observed in the literature e.g., (Hermansky *et al.*, 1983; Hermansky, 1990) that pitch frequency affects the estimation of the spectral envelope, in particular, the estimation of the spectral peaks, making the standard features sensitive to changes in pitch. Thus, we may expect a certain correlation between the standard cepstral features and pitch frequency (Singer and Sagayama, 1992; Fujinaga *et al.*, 2001). In a more recent work, the correlation between the average pitch frequency and vocal tract length has been used to obtain improved warp factors for vocal tract length normalization with a limited amount of speech data (Faria, 2003). There are different approaches proposed in the literature to extract pitch frequency (Hess, 1983). In our studies, the auxiliary feature pitch frequency was estimated using the simple inverse filter tracking (SIFT) algorithm (Markel, 1972). The speech signal was filtered by a low pass filter with a cutoff frequency of 800 Hz and down sampled to 2 kHz. Linear prediction analysis was then performed on this signal to extract LPC coefficients. The filtered speech signal was then passed through an inverse filter (defined by the LPC coefficients) to obtain the residual. An autocorrelation analysis was performed on the residual signal and the location of the second peak was taken as the pitch period, if the energy of the second peak was above a threshold (defined by the energy of the residual), otherwise a pitch frequency of 0 Hz is assigned. A five-point median smoothing was performed on the estimated pitch contour. We do not perform any further transformation as in (Fujinaga *et al.*, 2001) where its logarithmic form was used.

We evaluated our pitch estimator with speech from five males and five females (with a total duration of approximately five minutes) from the Keele pitch database (Plante *et al.*, 1995). The results of this evaluation in Table 4.6.1 show that the pitch estimation is reliable.

4.6.2 Rate-of-speech (ROS)

The effects of ROS can be observed in various aspects of speech recognition, including the duration model (transition probabilities), the acoustic model, and the pronunciation model. A change in ROS not only affects the duration of phones in the utterance, but at the same time it influences both the manner in which people articulate and the phonological variations of the words they produce (Hazen, 1998). In other words, changes in ROS could also affect the acoustic realization of

Gender	Voiced in error (%)	Unvoiced in error (%)	High gross error (%)	Low gross error (%)	AMD (Hz)
Female	6.5	2.9	1.1	16.0	3.7
Male	22.3	1.5	3.7	5.1	2.0

Table 4.1. Evaluation of pitch estimation algorithm for 5 male and 5 female utterances. Gross error = $\frac{n_c}{n_v}$, where n_c is the total number of comparisons for which the difference (absolute value) between the estimated pitch and the reference pitch is greater than 20% of the reference pitch and n_v is the total number of comparisons for which both the estimated pitch frequency and the reference pitch represent voiced speech. High gross error and low gross error basically represent the pitch doubling effect and pitch halving effect, respectively. AMD - Absolute mean deviation.

the phones. For instance, it has been observed that the formant frequencies ($F1, F2, F3$) of vowels differ significantly between slow, normal, and fast ROS. Furthermore, some vowels tend to be closer together on the $F1 - F2$ plane as the ROS increases, thus reflecting the neutralization of vowels (Kuwabara, 1997). The ROS can be measured in different ways, such as (Mirghafori *et al.*, 1995) used the hand labeled speech data, (Siegler, 1995) used forced aligned data to compute the speaking rate, (Samudravijaya *et al.*, 1998) and (Martinez *et al.*, 1998) used cepstral derivatives to compute speaking rate.

In our work, the ROS auxiliary feature was estimated using *mrate*[5] (Morgan and Fosler-Luisser, 1998). *mrate* incorporates multiple estimators:

1. Enrate: Enrate uses the first spectral moment of the wide band energy envelope as an estimate of ROS (Morgan *et al.*, 1997). The correlation between enrate and syllable rate is about 0.4.
2. Peak counting performed on the wide-band energy envelope.
3. The third estimator is sub-band-based, which computes a trajectory that is the average product of compressed sub-band energy trajectories. A peak counting algorithm is used above this to estimate rate-of-speech. The correlation between the estimated rate-of-speech and syllable rate is about 0.6.

The *mrate* is an average of the three measures described above. The *mrate* has been shown to be a good indicator of the syllable rate, having a correlation of 0.75 with the actual rate (Fosler-Lussier, 1999, Section 3.1.1).

4.6.3 Short-term energy

Energy is also known as an important prosodic attribute. It correlates with the stress property of vowels (Wang and Seneff, 2001) and with the syllabic structure (Nagarajan *et al.*, 2003); The short-term energy also has similarities to the pitch as the presence of non-zero pitch in the signal adds much energy to it. Unlike pitch and ROS, the short-term energy (log energy and its temporal derivatives) can often be found as a standard feature in normal ASR systems. Recently, proposed phase autocorrelation (PAC) based features obtained by energy normalization and an inverse cosine transform have shown robustness to noise in ASR but degraded performance in clean speech (Ikbal *et al.*, 2003b). However, it has been shown that the ASR performance in clean speech improves when short-term energy is used along with the PAC features (Ikbal *et al.*, 2003a). Energy can be computed in number of ways. In this thesis the auxiliary feature for energy was computed by taking the logarithm of the short-term energy of the windowed signal (using a Hamming window).

$$e(n) = \log(\frac{\sum_{i=1}^{i=M} s^2(i)}{M}) \tag{4.22}$$

[5] We would like to thank ICSI, Berkeley USA, for providing as the *mrate* software.

where $s(i)$ is the Hamming windowed speech signal and M is the size of the window. The energy contour computed as expressed above, in short-term follows the phoneme transitions and in long-term reflects the prosody.

In the following section, We first briefly summarize the observations from HMM/DBN-GMM ASR systems integrating the auxiliary features described above on isolated word recognition task and Numbers task. In Section 4.8, we present our studies on integrating the auxiliary features pitch frequency, short-term energy and ROS in hybrid HMM/ANN ASR system using context-independent phonemes as subword units.

4.7 Previous HMM/DBN-GMM Studies

In (Stephenson, 2003), the use of auxiliary features, such as, pitch frequency, short-term energy and ROS was investigated for a isolated word recognition task. The complete definition of this task can be found in Section 3.8.1 in Page 31. In the first set of experiments the MFCC features and auxiliary feature pitch frequency were discrete-valued. The codebook size of MFCC feature was 256 and for pitch frequency different codebook sizes were tried namely, 2, 4 and 8. A conventional baseline system, system "x_n, a_n (no assumption)" and system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" were trained and recognition studies were performed. This study yielded significant improvement in the performance for both systems "x_n, a_n (no assumption)" and "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" over the baseline system, when the auxiliary feature pitch frequency was hidden (Stephenson *et al.*, 2001). These studies were extended to continuous valued MFCC feature and pitch frequency where the MFCC distribution was modelled by a single Gaussian. Different systems, namely, Baseline, "$x_n \perp\!\!\!\perp a_n \,|\, q_n$", "$x_n, a_n$ (no assumption)" and "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" (BNs depicted in Figure 4.2) were trained and recognition studies were performed. The system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performed better than the baseline system while systems "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" and "$x_n, a_n$ (no assumption)" performed worse (Stephenson *et al.*, 2002). These studies were further extended to ASR systems with emission distribution modelled by GMMs (BNs depicted in Figure 4.2) integrating auxiliary features pitch frequency, short-term energy and ROS. Most of the systems trained with auxiliary features yielded performance similar to the baseline system. For systems trained with auxiliary feature pitch frequency systems "x_n, a_n (no assumption)" and "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" yielded improvement (though not statistically significant) when pitch frequency was observed (Stephenson, 2003, Table 6.7). For systems trained with auxiliary feature short-term energy, improvement (not statistically significant) was obtained for system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" when short-term energy was observed. Systems "x_n, a_n (no assumption)" and "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" performed worse when short-term energy was observed and yielded performance similar to baseline system when short-term energy was hidden (Stephenson, 2003, Table 6.8). Systems trained with auxiliary feature ROS yielded performance similar to the baseline system (Stephenson, 2003, Table 6.6). We present the hybrid HMM/ANN ASR studies on the same task in Section 4.8.1

The HMM/DBN-GMM studies were then extended to the Numbers task (connected word recognition). The complete definition of this task can be found in Section 3.8.2. The auxiliary features examined were pitch frequency, short-term energy and ROS. The training of the different HMM/DBN-GMM systems (BNs depicted in Figure 4.2) was performed on clean speech. The recognition studies were performed on both clean and noisy speech conditions. In clean condition, none of the systems integrating auxiliary features yielded improvement over the baseline system. In different noisy conditions, the systems (especially "$a_n \perp\!\!\!\perp q_n \,|\, x_n$") trained with auxiliary features (pitch frequency, short-term energy, ROS) yielded significant improvement over the baseline system when the auxiliary feature was hidden and its value was automatically inferred (Stephenson, 2003; Stephenson *et al.*, 2004). In our earlier hybrid HMM/ANN studies, we performed recognition studies in clean and noisy conditions. In noisy conditions, we found that HMM/ANN ASR systems integrating auxiliary features did not yield improvements similar to HMM/DBN-GMM systems over the baseline system (Stephenson *et al.*, 2004). In Section 4.8.2, we present the hybrid HMM/ANN ASR studies on the same task in clean speech conditions.

4.8 Hybrid HMM/ANN ASR System and Auxiliary Features

In this section, we present our studies on integrating auxiliary features in hybrid HMM/ANN ASR system with context-independent phonemes as subword units on different ASR tasks. The different tasks are isolated word recognition task and Numbers task. For more details about these tasks refer to Section 3.8.1 and Section 3.8.2, respectively.

For both the tasks, we studied different hybrid HMM/ANN systems trained with standard features (x_n) and different auxiliary features (a_n):

- System "Baseline": Hybrid HMM/ANN system trained only with x_n.
- System "$x_n \perp\!\!\!\perp a_n \,|\, q_n$": Two separate ANNs were trained, one with x_n as the input feature and the other with a_n as the input feature. In case of auxiliary feature pitch frequency, a_n was normalized by the highest pitch frequency that can be estimated by the pitch estimation algorithm (i.e., 400Hz). This was done in order to avoid saturation of the activation function (e.g., sigmoids) (LeCun *et al.*, 1998). The auxiliary feature a_n is always observed (both during training and recognition).
- System "x_n, a_n (no assumption)": A single ANN was trained. The input feature was the augmented feature vector (x_n, a_n). Similar to system "$x_n \perp\!\!\!\perp a_n \,|\, q_n$", for the auxiliary feature pitch frequency, a_n was normalized by the highest pitch frequency that can be estimated by the pitch estimation algorithm. The auxiliary feature a_n is always observed (both during training and recognition).
- System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$": The auxiliary feature of the training data was quantized into three regions and three ANNs corresponding to the three regions were trained. When performing recognition studies, we had two cases:
  1. a_n observed: The output of the ANN corresponding to the value of a_n is used to estimate the local likelihood.
  2. a_n hidden: a_n is hidden according to (4.11) and, the resulting local likelihood is used for decoding.

The number of parameters for all the systems described above were same for both the tasks.

In the following subsections, we present the results of the recognition experiments on both the tasks.

4.8.1 Isolated Word Recognition Task

We use the PhoneBook speech corpus for speaker-independent task-independent, small vocabulary (75 words) isolated word recognition task. The acoustic vector is 21 dimensional MFCCs as done in (Zweig, 1998). The details of the experimental setup are described in Section 3.8.1. The test set contains eight different list of words. The performance is the average of the eight word error rates corresponding to the eight different lists in the test set.

The different auxiliary features for this ASR task were estimated in the following way:

1. Pitch frequency was estimated using SIFT algorithm (as described earlier in Section 4.6.1) with a frame size and shift of 25 ms and 8.3 ms, respectively.
2. Short-term energy was estimated (as described earlier in Section 4.6.3) with a frame size and shift of 25 ms and 8.3 ms, respectively.
3. ROS was computed for the whole utterance by *mrate* software and, then it was repeated every 8.3 ms. This was done because the isolated words are of shorter duration.

The results of the recognition experiments are given in Table 4.2. The observations for each auxiliary feature is summarized as follows:

- Pitch frequency: Systems "x_n, a_n (no assumptions)", "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" (when observed) and "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" perform significantly better than the "Baseline" system. It is interesting to note that system "x_n, a_n (no assumptions)" where the pitch frequency is treated as an additional feature yields the best performance. This is contrary to the previous studies integrating pitch frequency in HMM/GMM systems (Fujinaga *et al.*, 2001), where it has been observed that using the pitch frequency as an additional feature degrades the performance of the ASR system. The main reason behind this difference is the ability of the ANN to model higher order correlation between x_n and a_n through the hidden layer (Bourlard and Morgan, 1994).

- Short-term Energy: System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" yields significant improvement over the "Baseline" line system. Systems "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" and "$x_n, a_n$ (no assumption)" perform significantly worse than the system "Baseline".

- ROS: Systems "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" ($a_n$ hidden) and "x_n, a_n (no assumption)" perform better than the baseline system. However, the improvement is statistically significant only for system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" when a_n is hidden. The reason behind this can be that ROS has very less impact on short utterance. Also, since the utterances are of very short duration the estimate of ROS may not be reliable.

Though, system "x_n, a_n (no assumption)" for auxiliary feature pitch frequency yields the best performance, system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs better than the baseline system for all the auxiliary features (except when ROS is observed).

Systems	O or H	Pitch	Energy	ROS
Baseline	-	4.7		
$x_n \perp\!\!\!\perp a_n \,\|\, q_n$	O	$4.1^{\dagger}$	5.1	4.8
x_n, a_n (no assumption)	O	$\mathbf{2.8}^{\dagger}$	5.6	4.5
$a_n \perp\!\!\!\perp q_n \,\|\, x_n$	O	$3.7^{\dagger}$	$\mathbf{3.4}^{\dagger}$	4.8
	H	4.4	$3.9^{\dagger}$	$\mathbf{3.7}^{\dagger}$

Table 4.2. Word error rate expressed in percentage for the Baseline hybrid HMM/ANN system and for the hybrid HMM/ANN systems integrating auxiliary features on isolated word recognition task. The system in Figure 4.4 (a) was used for Baseline. The system in Figure 4.4 (b) was used for $x_n \perp\!\!\!\perp a_n \,|\, q_n$. The system in Figure 4.4 (c) was used for x_n, a_n (no assumption). The system in Figure 4.4 (d) for $a_n \perp\!\!\!\perp q_n \,|\, x_n$. Notations: O means auxiliary feature pitch frequency is observed and H means the auxiliary feature is hidden. The performances of the hybrid HMM/ANN systems integrating pitch frequency are given in the third column. The performances of the hybrid HMM/ANN systems integrating short-term energy are given in the fourth column. The performances of the hybrid HMM/ANN systems integrating ROS are given in the fifth column. Bold face indicates the best system in each column. † indicates that the improvement in the performance of the system (over baseline) is statistically significant (95% confidence level or above).

4.8.2 Numbers task

The OGI-Numbers95 database containing spontaneously spoken free-format numbers over telephone channel has been used for this study. Following the past and present studies on Numbers task at IDIAP (Hagen, 2001), the acoustic vector x_n is the 39 dimensional PLP cepstral coefficients extracted from the speech signal using a window of 25 ms with a shift of 12.5 ms. Further details about the task can be found in Section 3.8.2.

The different auxiliary features for this study were estimated in the following way:

1. The pitch frequency was estimated by SIFT algorithm with frame size and shift of 25 ms and 12.5 ms, respectively (as described in Section 4.6.1).

2. Short-term energy was estimated with a frame size and shift of 25 ms and 12.5 ms, respectively.

3. The ROS was computed every 12.5 ms using *mrate* software with a window size of 1 s.

The results of the recognition experiments are given in Table 4.3. The observations for each auxiliary feature is summarized as follows:

- Pitch frequency: System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs significantly better than the baseline system when the auxiliary feature pitch frequency is observed. Hiding the auxiliary feature hurts the performance of the system. Systems "$x_n \perp\!\!\!\perp a_n \,|\, q_n$", "$x_n, a_n$ (no assumption)" and "Baseline" yield statistically similar performance. It can be observed from the results that system "x_n, a_n (no assumption)" does not yield improvement similar to isolated word recognition task. The possible reasons for this is that the PhoneBook database has more phonetic variation and phoneme classes compared to OGI Numbers95 database.

- Short-term energy: System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs significantly better than the baseline system when the auxiliary feature energy is observed. Like in the case of pitch frequency, the performance of the system drops significantly when the auxiliary feature is hidden. Systems "$x_n \perp\!\!\!\perp a_n \,|\, q_n$", "$x_n, a_n$ (no assumption)" and "Baseline" yield statistically similar performance.

- ROS: System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs better than the baseline system when ROS is observed. However, this improvement is not statistically significant. Systems "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" and "Baseline" yield statistically similar performance. System "x_n, a_n (no assumption)" performs significantly worse than the baseline system.

Overall, the hybrid HMM/ANN system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs better than the system "Baseline" for all auxiliary features, when the auxiliary feature is observed.

Systems	O or H	Pitch	Energy	ROS
Baseline	-	8.7		
$x_n \perp\!\!\!\perp a_n \,|\, q_n$	O	8.7	8.6	8.7
x_n, a_n (no assumption)	O	8.6	8.8	9.3
$a_n \perp\!\!\!\perp q_n \,|\, x_n$	O	**8.0**†	**7.5**†	**8.4**
	H	9.5	9.4	8.8

Table 4.3. Word error rate expressed in percentage for the Baseline hybrid HMM/ANN system and for the hybrid HMM/ANN systems integrating auxiliary features on Numbers task. The system in Figure 4.4 (a) was used for Baseline. The system in Figure 4.4 (b) was used for $x_n \perp\!\!\!\perp a_n \,|\, q_n$. The system in Figure 4.4 (c) was used for x_n, a_n (no assumption). The system in Figure 4.4 (d) for $a_n \perp\!\!\!\perp q_n \,|\, x_n$. Notations: O means auxiliary feature pitch frequency is observed and H means the auxiliary feature is hidden. The performances of the hybrid HMM/ANN systems integrating pitch frequency are given in the third column. The performances of the hybrid HMM/ANN systems integrating short-term energy are given in the fourth column. The performances of the hybrid HMM/ANN systems integrating ROS are given in the fifth column. Bold face indicates the best system in each column. † indicates that the improvement in the performance of the system (over baseline) is statistically significant (95% confidence level or above).

4.9 Discussion

In the previous section, we presented the ASR studies on hybrid HMM/ANN systems integrating auxiliary features pitch frequency, short-term energy and ROS on two different tasks. The best performance on the two tasks are:

1. Isolated word recognition task: 2.8% WER achieved by using auxiliary feature pitch frequency as an additional feature.

2. Numbers task: 7.5% WER achieved by conditioning the emission distribution upon the auxiliary feature short-term energy.

In both the tasks we observe that,

- Conditioning the emission distribution upon the auxiliary feature leads to improvement in the ASR performance.
- ROS has the least influence on the performance of the ASR.

Random Auxiliary Features

It can be hypothesized that using auxiliary feature provides increased model flexibility and, this can result in improved performance. Hence, in order to verify that the improvement in the performance of the ASR systems using auxiliary features is not due to increased model flexibility, we perfomed experiments using random auxiliary features (Magimai.-Doss *et al.*, 2003a). The experimental studies showed that the performance of the ASR system degrades or remains closer to the baseline system when using random auxiliary features. In other words, the improvement in the performance of ASR system integrating auxiliary feature is due to the extra acoustic information that the auxiliary feature provides.

4.10 Summary and Conclusion

In this chapter, we introduced the notion of auxiliary feature. We presented different approaches to integrate auxiliary features in state-of-the-art HMM-based ASR system. The two main approaches of interest are:

- Treating auxiliary feature as an additional feature by concatenating them to the standard feature.
- Conditioning the emission distribution upon the auxiliary feature.

The auxiliary features that were investigated are (1) pitch frequency, (2) short-term energy and (3) rate-of-speech. These features were directly extracted from the speech signal.

The proposed approaches to integrate auxiliary features were studied in the framework of hybrid HMM/ANN ASR system with context-independent phonemes as subword units. The main conclusions of the ASR studies are the following:

- Performance of the standard ASR system can be improved by integrating auxiliary features.
- It is better to use auxiliary features to condition the emission distribution rather than using them as additional features.

Chapter 5

Auxiliary Features for CD Phoneme-Based ASR

5.1 Introduction

In the previous chapter, we presented different ways to integrate auxiliary features in standard HMM-based ASR systems. ASR studies conducted on two different tasks, isolated word recognition task and Numbers task, showed that it is better to condition the acoustic models upon the auxiliary feature pitch frequency, short-term energy and ROS. These ASR studies were done with ASR systems using context-independent phonemes as subword units. State-of-the-art HMM-based ASR systems use context-dependent (CD) phonemes as subword units in order to handle the coarticulation effects (Schwartz *et al.*, 1985).

In this chapter, we present our studies on integrating auxiliary features (1) pitch frequency, (2) short-term energy and (3) ROS in ASR systems with context-dependent phonemes as subword units. We have studied this on two different tasks, namely, Numbers task and conversational telephone speech (CTS) task. Section 5.2 presents the ASR studies conducted on Numbers task in the framework of hybrid HMM/ANN system, HMM/DBN-GMM system and TANDEM system. In Section 5.3, we present the preliminary studies on CTS task. Section 5.4 presents a short discussion based on the experimental studies. We finally summarize and conclude in Section 5.5

5.2 Numbers Task

We performed ASR studies with context-dependent phonemes on OGI Numbers 95 database within the framework of HMM/DBN-GMM, hybrid HMM/ANN and TANDEM systems. The OGI Numbers95 database contains 80 context-dependent phonemes. For more details about the experimental setup of Numbers task refer to Section 3.8.2. The standard feature (x_n) is the 39 dimensional PLP feature vector and the auxiliary features (a_n) are pitch frequency, short-term energy and ROS. The same features were used earlier for the ASR studies using context-independent phonemes in Chapter 4.

5.2.1 Hybrid HMM/ANN

For each of the auxiliary feature, we trained different hybrid HMM/ANN systems shown in Figure 4.4:

- System "Baseline": Standard hybrid HMM/ANN system trained only with x_n.

- System "$x_n \perp\!\!\!\perp a_n \,|\, q_n$": x_n and a_n are modelled by separate ANNs i.e., x_n and a_n are treated as separate streams of information.
- System "x_n, a_n (no assumption)": a_n is treated as an additional feature i.e., concatenated with x_n.
- System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$": a_n conditions the emission distribution. a_n is discrete valued and, the values of a_n are equal to the one used in our earlier ASR studies (with context-independent phonemes) presented in Section 4.8.2. For each of the values of a_n an ANN is trained. During recognition, we have two cases:
  - a_n observed: The output of the ANN corresponding to the value of a_n is used to estimate the local likelihood.
  - a_n hidden: a_n is hidden according to (4.11) and the resulting local likelihood is used for decoding.

The detailed explanation about the implementation of the different systems described above can be found in Section 4.5.2. In case of system "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" and system "$x_n, a_n$ (no assumption)" with pitch frequency as the auxiliary feature, the pitch frequency was normalized by the highest frequency that could be estimated by the pitch estimation algorithm. All the systems had the same number of parameters and they were equal to that of the systems earlier trained with context-independent phonemes. The results of integrating the auxiliary features pitch frequency, short-term energy and ROS in hybrid HMM/ANN system are presented in Table 5.1 along with the baseline performance.

Systems	O or H	Pitch	Energy	ROS
Baseline	-	6.8		
$x_n \perp\!\!\!\perp a_n \,\|\, q_n$	O	7.0	7.3	7.3
x_n, a_n (no assumption)	O	**6.2**†	**6.3**†	**6.3**
$a_n \perp\!\!\!\perp q_n \,\|\, x_n$	O	7.2	7.4	7.8
	H	8.1	7.6	8.1

Table 5.1. Word error rate expressed in percentage for the Baseline and for the hybrid HMM/ANN systems integrating auxiliary feature on Numbers task. The subword units are context-dependent phonemes. The system in Figure 4.4 (a) was used for Baseline. The system in Figure 4.4 (b) was used for $x_n \perp\!\!\!\perp a_n \,|\, q_n$. The system in Figure 4.4 (c) was used for x_n, a_n (no assumption). The system in Figure 4.4 (d) for $a_n \perp\!\!\!\perp q_n \,|\, x_n$. Notations: x_n = PLP feature, a_n = auxiliary feature (Pitch - pitch frequency, Energy - short-term energy, ROS - rate-of-speech), q_n = HMM state, O means auxiliary feature a_n is observed and H means auxiliary feature a_n is hidden. Bold face indicates the best system for each auxiliary feature (performing better than the baseline system). † indicates that the improvement in the performance of the system is statistically significant (95% confidence level or above).

We observe that system "x_n, a_n (no assumption)" performs better than the system "Baseline" for all the auxiliary features. System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" and system "$x_n \perp\!\!\!\perp a_n \,|\, q_n$" perform worse than the baseline for all auxiliary features. These results are contrary to the results obtained in the previous chapter on the same ASR task with context-independent phonemes as subword units, where, system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" performs the best for all the auxiliary features (when observed) and, system "x_n, a_n (no assumption)" performs similar to the system "Baseline" (see Table 4.3).

5.2.2 HMM/DBN-GMM

We use the DBN software developed in (Stephenson, 2003) to train ASR systems with 80 context-dependent phonemes, 3 emitting states per phoneme and 12 mixtures per state. Results using the auxiliary features pitch frequency, short-term energy and ROS in HMM/DBN-GMM system are given in Tables 5.2 along with the baseline performance. The different systems are:

- System "Baseline": This system is only trained with x_n i.e., equivalent to standard HMM-based system.

- System "$x_n \perp\!\!\!\perp a_n \,|\, q_n$": a_n is treated as an additional feature i.e., concatenated with x_n.
- System "x_n, a_n (no assumption)": a_n is treated as a mid level feature i.e., it conditions the emission distribution and, at the same time it is treated as an additional feature.
- System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$": a_n conditions the emission distribution.

The details about the implementation of all the above described systems can be found in 4.5.1. System "x_n, a_n (no assumption)" and system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" have conditional Gaussians, while all other have systems have Gaussians. During recognition, the auxiliary feature is hidden according to (4.21).

Unlike the hybrid HMM/ANN systems, all the HMM/DBN-GMM systems trained with standard feature and auxiliary features perform worse than the baseline system (except system "$x_n \perp\!\!\!\perp a_n \,|\, \{q_n, J_n\}$" for auxiliary feature ROS). The performance of system "$x_n \perp\!\!\!\perp a_n \,|\, \{q_n, J_n\}$" where the auxiliary feature is treated as additional feature is statistically similar to the baseline system for all auxiliary features.

Systems	O or H	Pitch	Energy	ROS
Baseline	-	7.3		
$x_n \perp\!\!\!\perp a_n \,\|\, \{q_n, J_n\}$	O	7.8	7.4	**7.1**
$x_n \perp\!\!\!\perp a_n \,\|\, \{q_n, J_n\}$	H	7.9	7.7	7.5
x_n, a_n (no assumption)	O	9.9	8.6	7.9
x_n, a_n (no assumption)	H	9.5	7.7	7.8
$a_n \perp\!\!\!\perp q_n \,\|\, x_n$	O	8.5	9.2	8.6
$a_n \perp\!\!\!\perp q_n \,\|\, x_n$	H	8.1	8.5	7.7

Table 5.2. (Word error rate expressed in percentage for the Baseline HMM/DBN-GMM system and for the HMM/DBN-GMM systems integrating auxiliary features Numbers task. The subword units are context-dependent phonemes. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (a) was used for Baseline. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (b) was used for $x_n \perp\!\!\!\perp a_n \,|\, q_n$. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (c) was used for x_n, a_n (no assumption). The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (d) for $a_n \perp\!\!\!\perp q_n \,|\, x_n$. Notations: x_n = PLP features, a_n = auxiliary feature (Pitch - pitch frequency, Energy - short-term energy, ROS - rate-of-speech), q_n = discrete states, O means a_n is observed and H means a_n is hidden. Bold face indicates the best system for each auxiliary feature (performing better than the baseline system).

5.2.3 TANDEM

In this section, we present ASR studies integrating auxiliary features using TANDEM systems (Magimai.-Doss *et al.*, 2004b). As described in Section 4.5.3, there are two ways to integrate auxiliary feature in TANDEM systems, (a) integrating auxiliary features before tandem feature extraction ("Tandem(CEP+AUX)") and (b) modelling auxiliary features along with tandem features ("Tandem(CEP)+AUX"). In rest of this section, we present the ASR studies.

Tandem(CEP+AUX)

"Tandem(CEP+AUX)" approach integrates the auxiliary feature through ANN, i.e., before the extraction of tandem-features. Figure 4.5 gives a description of this approach. We studied different systems:

- A HMM/GMM baseline system with PLP features.
- *Tandem(CEP)*: TANDEM baseline system trained with the tandem features extracted from the ANN of the hybrid HMM/ANN system "Baseline" with context-independent phonemes as subword units (earlier used in the studies presented in Section 4.8.2).

- *Tandem(CEP+AUX-A)*: Tandem system integrating the auxiliary feature, where, the tandem-features for each of the auxiliary feature pitch frequency, short-term energy and ROS were extracted from the ANN of their respective "x_n, a_n (no assumptions)" hybrid HMM/ANN system with context-independent phonemes as subword units.

- *Tandem(CEP+AUX-C)*: Tandem systems integrating the auxiliary features, where, for each auxiliary feature pitch frequency, short-term energy and ROS, the tandem features were extracted from the ANN of their respective "$a_n \perp\!\!\!\perp q_n \mid x_n$" hybrid HMM/ANN system with context-independent phonemes as subword units. In this system we have two cases:
  - Auxiliary feature observed: When the auxiliary feature is observed, at each time frame n the output of the MLP corresponding to the discrete auxiliary feature is selected. The tandem-features are then extracted by transforming the resulting posteriors.
  - Auxiliary feature hidden: In the case where the auxiliary feature is hidden, the posteriors resulting after marginalizing out auxiliary feature[1] are transformed into tandem-features.

We used the HTK-toolkit (Young *et al.*, 1997) to train the HMM/GMM system with 80 context-dependent phonemes, 3 emitting states per phoneme and 12 mixtures per state. All the systems were trained with clean data. During recognition, apart from testing our system on clean data, we also tested our systems on versions with added noise using the Noisex-92 database (Varga *et al.*, 1992). Non-stationary factory (FACT) and stationary helicopter (LYNX) noise conditions at signal-to-noise ratios (SNR) 6dB and 12dB were studied. The results of the recognition studies are given in Table 5.3. It can be seen that the TANDEM systems perform better than the baseline system using PLP features in both clean and noisy conditions. In clean conditions, *Tandem(CEP+AUX-A)* for auxiliary feature ROS performs better than the *Tandem(CEP)*. The performance of *Tandem(CEP+AUX-A)* for auxiliary features short-term energy and ROS degrades significantly in noisy conditions. The main reason for this is that the estimation of auxiliary features is not reliable. One solution would be to hide the continuous valued a_n, but it is not obvious how this could be done in the case of hybrid HMM/ANN systems. In the context-independent phoneme studies, we observed that System "$a_n \perp\!\!\!\perp q_n \mid x_n$" yielded better performance compared to the baseline system when auxiliary feature was observed (see Table 4.3), but *Tandem(CEP+AUX-C)* performs worse than the *Tandem(CEP)* for all the auxiliary features. Similar trend has been observed earlier in literature (Ellis *et al.*, 2001), where the improvements in the context-independent system does not shows up in the context-dependent HMM-GMM system using tandem features. In our case, the reason for this could be the switching between the ANNs corresponding to the observed discrete-valued auxiliary feature. Since the different ANNs model different distributions, the switching between them may be affecting the PCA analysis part of the tandem-feature extraction.

Tandem(CEP)+AUX

In "Tandem(CEP)+AUX", the tandem feature is extracted from ANN trained only with standard feature. The tandem-feature is then modelled along with the auxiliary features. Figure 4.6 gives a description of this approach.

We trained different systems:

- A HMM/DBN-GMM system with PLP features.

- *Tandem(CEP)*: Similar to Tandem(CEP+AUX) approach, we trained a HMM/DBN-GMM TANDEM baseline system *Tandem(CEP)* with the tandem features $Z = \{z_1, \cdots z_n, \cdot z_N\}$ extracted from the ANN of the hybrid HMM/ANN system "Baseline" with context-independent phonemes as subword units(used for studies reported in Section 4.8.2).

[1] $p(q_n|x_n) = \sum_{l=1}^{L} p(q_n|x_n, a_n = l) \cdot p(a_n = l)$

	AF	O or H	∞	LYNX 12	LYNX 6	FACT 12	FACT 6
PLP	-	-	7.3	11.6	20.0	16.2	37.6
Tandem(CEP)	-	-	4.9	9.4	**16.2**	**13.2**	**25.6**
Tandem(CEP+AUX-A)	Pitch	O	5.1	**9.1**	16.3	13.8	26.2
Tandem(CEP+AUX-C)	Pitch	O	5.5	9.9	16.4	14.6	31.2
	Pitch	H	5.4	9.4	17.0	14.5	28.0
Tandem(CEP+AUX-A)	Energy	O	5.7	19.3	46.6	34.0	70.1
Tandem(CEP+AUX-C)	Energy	O	5.7	10.9	19.3	15.8	28.8
	Energy	H	5.6	10.1	17.2	14.5	28.4
Tandem(CEP+AUX-A)	ROS	O	**4.8**	15.0	34.6	26.3	59.0
Tandem(CEP+AUX-C)	ROS	O	6.0	10.7	18.1	15.9	30.8
	ROS	H	6.0	10.5	17.8	15.4	28.9

Table 5.3. Results of Tandem(CEP+AUX) approach on Numbers task where the tandem-features are extracted from hybrid HMM/ANN system modelling PLP features and auxiliary features. Results are reported for clean data (SNR=∞), SNRs of 6dB and 12dB. The performance is measured in-terms of word error rate (expressed in %). The best system for each condition is marked in boldface. Notations: AF - auxiliary features, Pitch - pitch frequency, Energy - short-term energy, ROS - rate-of-Speech, O - auxiliary feature is observed and H - auxiliary feature is hidden.

- *Tandem(CEP)+AUX-A*: HMM/DBN-GMM system "$z_n \perp\!\!\!\perp a_n \mid \{q_n, J_n\}$" corresponding to the BN in Figure 4.2 (b) with tandem feature Z and auxiliary features pitch frequency, short-term energy and ROS (auxiliary feature is used as an additional feature).

- *Tandem(CEP)+AUX-C*: HMM/DBN-GMM system "$a_n \perp\!\!\!\perp q_n \mid z_n$" corresponding to the BN in Figure 4.2 (d) with tandem features Z and auxiliary features pitch frequency, short-term energy and ROS (emission distribution is conditioned upon the auxiliary feature).

We use the DBN software developed in (Stephenson, 2003) to train ASR systems with 80 context-dependent phonemes, 3 emitting states per phoneme and 12 mixtures per state. We performed recognition studies on both clean and noisy conditions. The results of the recognition studies are given in Table 5.4. During recognition, the auxiliary feature was hidden according to (4.21). The TANDEM systems again perform better than the PLP baseline system in both clean and noisy conditions. When comparing between TANDEM systems in clean condition the system *Tandem(CEP)+AUX-A* performs better than the system *Tandem(CEP)*. In order to verify that this improvement is not due to an increase in the number of parameters, we trained a *Tandem(CEP)* system with 18 mixtures. The performance of this system is 5.1% in clean, 8.8% (LYNX SNR 12dB), 15.4% (LYNX SNR 6dB), 12.3% (FACT SNR 12dB) and 24.6% (FACT SNR 6dB). *Tandem(CEP)+AUX-A* performs better than this system in all conditions when the auxiliary features short-term energy and ROS are hidden. It can be seen from the results that *Tandem(CEP)+AUX-A* system yields similar performance for all the auxiliary features. The main reason for this can be that the auxiliary features are providing the same kind of additional information. For instance, pitch frequency and short-term energy both provide voicing information.

5.2.4 Short Summary

In this section, we studied context-dependent phoneme-based ASR systems integrating auxiliary features in the framework of hybrid HMM/ANN systems, HMM/DBN-GMM systems and TANDEM systems. These studies show that:

- Appending the auxiliary feature to standard feature improves the performance of ASR system. However, the improvement is not always statistically significant.

				LYNX		FACT	
	AF	O or H	∞	12	6	12	6
PLP	-	-	7.3	16.3	33.3	24.6	46.9
Tandem(CEP)	-	-	5.2	9.3	15.4	13.0	24.6
Tandem(CEP)+AUX-A	Pitch	O	4.9	8.3†	14.6	12.5	25.0
	Pitch	H	4.9	8.6	15.3	12.3	24.7
Tandem(CEP)+AUX-C	Pitch	O	5.4	9.6	15.9	13.4	25.5
	Pitch	H	5.8	9.6	16.1	13.0	24.7
Tandem(CEP)+AUX-A	Energy	O	4.9	8.4	14.8	12.6	24.2
	Energy	H	4.8	**8.2**†	15.0	**11.9**†	**23.7**
Tandem(CEP)+AUX-C	Energy	O	6.1	10.6	17.7	13.8	25.5
	Energy	H	5.5	9.6	17.0	13.6	24.8
Tandem(CEP)+AUX-A	ROS	O	**4.7**†	**8.2**†	14.8	12.8	26.6
	ROS	H	4.8	**8.2**†	**14.3**	12.3	24.2
Tandem(CEP)+AUX-C	ROS	O	5.7	10.0	16.7	13.7	25.6
	ROS	H	5.6	9.8	16.3	13.4	25.4

Table 5.4. Results of Tandem(CEP)+AUX approach on Numbers task where the tandem-features are extracted from a hybrid HMM/ANN baseline system and, are modelled along with auxiliary features using DBNs (HMM/DBN-GMM). For systems using auxiliary features, the first row corresponds to the case when the auxiliary features are observed and the second row to the case when the auxiliary features are hidden. Results are reported for clean data (SNR=∞), SNRs of 6dB and 12dB. The performance is measured in-terms of word error rate (expressed in %). † Systems performing signficantly better than *Tandem(CEP)* system. The best system(s) for each condition is marked boldface. Notations: AF - auxiliary features, Pitch - pitch frequency, Energy - short-term energy, ROS - rate-of-Speech, O - auxiliary feature is observed and H - auxiliary feature is hidden.

- TANDEM system performs better than the standard cepstral features based HMM/GMM system in both clean and noisy conditions. The performance of the TANDEM system can be further improved by integrating auxiliary features.

- In the framework of TANDEM system, "Tandem(CEP)+AUX" is the better approach (compared to "Tandem(CEP+AUX)" approach) to integrate auxiliary features.

- The best performance of 4.7% word error rate in clean condition was achieved with system *Tandem(CEP)+AUX-A* when auxiliary feature ROS was observed.

5.3 Conversational Telephone Speech Task

The experimental studies until now have focussed upon small tasks such as isolated word recognition (on PhoneBook database) and connected word recognition (on OGI Numbers95 database). However, ASR research is more and more focusing towards recognizing conversational speech, where there is lot of variability in the speech signal. Hence, we extended our efforts to integrate auxiliary feature in standard ASR to conversational telephone speech recognition task.

The acoustic feature used for this task is 39 dimensional PLP cepstral coefficients. The PLP features were computed with vocal tract normalization. Utterance level mean and variance normalization was performed on the PLP cepstral coefficients. The auxiliary features pitch frequency, ROS and short-term energy were extracted as described in Sections 4.6.1, 4.6.2 and 4.6.3. There are 46 context-independent phonemes. The lexicon consists of 1000 words with multi-words and multi-pronunciations. The language model is a bi-gram model. For further details about the CTS task refer to Section 3.8.3.

We trained gender dependent HMM/GMM systems without using auxiliary features:

- PLP Baseline: For each gender type, a baseline HMM/GMM system was trained with PLP features.

- *Tandem(CEP)*: A TANDEM system was trained for each gender type with 46 dimensional tandem-features. The tandem-features were obtained from their respective gender-dependent ANNs. The gender dependent ANNs (with 9 frames of PLP acoustic features as input) were trained with 14.6 hours of training data, while the remaining 1.4 hours of training data was used as cross-validation set to prevent over-training.

The gender dependent HMM/GMM system were trained through 40 iterations: 5 iterations for the context-independent models, 5 iterations for the context-dependent models, 5 iterations for the clustered context-dependent models, and then 5 iteration each for incrementing mixtures from 1 to 32 (2, 4, 8, 16, 32) using HTK toolkit (Young *et al.*, 1997).

Table 5.5 presents the results of the PLP baseline system and the TANDEM system. The TANDEM system performs better than the HMM/GMM PLP baseline system.

We take the "Tandem(CEP+AUX)" approach in the framework of TANDEM systems to integrate auxiliary features[2]. A description of this approach is given in Section 4.5.3. In this approach, the auxiliary feature is first integrated in hybrid HMM/ANN system and then, the ANNs trained with standard feature and auxiliary feature are used to extract tandem-features. We trained different hybrid HMM/ANN systems integrating auxiliary features system "x_n, a_n (no assumptions)" (auxiliary feature is treated as an additional feature) and system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" (auxiliary feature conditions the emission distribution). In case of hybrid HMM/ANN system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" integrating auxiliary feature, for each auxiliary feature and for each gender, the auxiliary feature was quantized into three regions and ANN for each region was trained. We then trained TANDEM systems with tandem-features obtained from the ANNs trained with standard feature and auxiliary feature:

1. *Tandem(CEP+AUX-A)*: The tandem-feature is extracted from the ANN of the hybrid HMM/ANN system "x_n, a_n (no assumptions)".

2. *Tandem(CEP+AUX-C)*: The tandem-feature is extracted from the ANN of the hybrid HMM/ANN system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$". Similar to the Numbers task, we study two cases:

    - Auxiliary feature observed: At each time frame n the output of the MLP corresponding to the discrete auxiliary feature is selected. The tandem-features are then extracted by transforming the resulting posteriors.

    - Auxiliary feature hidden: At each time frame, the posteriors resulting after marginalizing out the auxiliary feature are transformed into tandem-features.

As observed earlier on Numbers task, the TANDEM system performs better than the HMM/GMM PLP baseline system. System *Tandem(CEP+AUX-A)* for auxiliary features pitch frequency and short-term energy performs similar to the *Tandem(CEP)* system. System *Tandem(CEP+AUX-C)* performs worse than system *Tandem(CEP)* for all the auxiliary features. Similar trend was observed in the TANDEM studies on Numbers task.

In (Zhu *et al.*, 2004), it was shown that standard PLP features and tandem-features carry complimentary information. Thus, by combining PLP features and tandem-features at feature level, the performance of the ASR system can be improved. In (Zhu *et al.*, 2004), it was found that by concatenating the 39 dimensional PLP feature vector to the tandem-feature vector truncated to first 25 dimensions yields significant improvement over the PLP baseline system. It was also observed that keeping all the dimensions does not help always and, truncating the tandem-feature vector to below first 15 dimensions hurts the performance of ASR. We performed ASR studies by training different

[2]The "Tandem(CEP)+AUX" approach was not investigated for the CTS task due to limits of the DBN software.

System	AF	O or H	Male	Female
PLP Baseline	-	-	43.9	40.6
Tandem(CEP)	-	-	**42.2**	39.6
Tandem(CEP+AUX-A)	Pitch	O	42.3	39.5
Tandem(CEP+AUX-C)	Pitch	O	45.3	41.1
	Pitch	H	43.6	41.0
Tandem(CEP+AUX-A)	Energy	O	42.3	**39.3**
Tandem(CEP+AUX-C)	Energy	O	42.7	40.9
	Energy	H	43.0	40.3
Tandem(CEP+AUX-A)	ROS	O	43.3	41.2
Tandem(CEP+AUX-C)	ROS	O	45.3	43.3
	ROS	H	44.9	42.7

Table 5.5. Results of continuous speech recognition studies using auxiliary features. The performance is measured in-terms of word error rate (expressed in %). AF denotes auxiliary features, O denotes the auxiliary feature is observed and H denotes the auxiliary feature is hidden. Pitch - Pitch frequency, Energy - Short-term energy and ROS - Rate-of-speech. Bold face indicates the best system.

TANDEM systems concatenating the 39 dimensional PLP feature vector and the tandem-feature vector truncated to first 17 dimensions[3]:

- PLP+TF(CEP): The truncated tandem features are obtained from the baseline gender dependent ANN.
- PLP+TF(CEP+AUX-A): For each auxiliary feature, the truncated tandem features are obtained from the ANN of hybrid HMM/ANN system "x_n, a_n (no assumptions)".
- PLP+TF(CEP+AUX-C): For each auxiliary feature, the truncated tandem features are obtained from the ANNs of hybrid HMM/ANN system "$a_n \perp\!\!\!\perp q_n \,|\, x_n$".

The results of the recognition studies are given in Table 5.6. The concatenation of PLP and tandem features performs better than PLP baseline and their respective TANDEM systems (see Table 5.5). System PLP+TF(CEP) performs the best. The systems with auxiliary feature that performs closer to PLP+TF(CEP) is system PLP+TF(CEP+AUX-A) and system PLP+TF(CEP+AUX-C) for auxiliary feature short-term energy. In the CTS studies, we have until now seen that integrating auxiliary features pitch frequency, short-term energy and ROS does not yield any improvement in the performance of ASR like the isolated word recognition task and Numbers task.

5.4 Discussion

In this chapter and the previous chapter, we studied the integration of auxiliary features in state-of-the-art ASR systems with context-independent phoneme as subword units and context-dependent phoneme as subword units. We observed that integration of the auxiliary features pitch frequency, short-term energy and ROS can improve the performance of the system. The manner in which these auxiliary features are integrated into the system can dramatically influence the performance. In systems with context-independent phonemes, we observed that conditioning the acoustic models upon the auxiliary feature helps in improving the performance of ASR. However, in systems with context-dependent phonemes, this trend is not observed. The systems trained with standard features concatenated with auxiliary features perform better.

When the acoustic models are conditioned upon the auxiliary feature, the key idea is to make the acoustic models robust to the variabilities present in the speech signal through the integration

[3] In (Zhu *et al.*, 2004), it was observed that the first 17 dimensions and the first 25 dimensions covered 95% and 98% of total variance, respectively. Since we are studying a system which is similar to (Zhu *et al.*, 2004) configuration we chose 17 dimensions.

System	AF	O or H	Male	Female
PLP Baseline	-	-	44.0	40.6
Tandem(CEP)	-	-	42.2	39.6
PLP+TF(CEP)	-	-	**40.7**	**37.7**
PLP+TF(CEP+AUX-A)	Pitch	O	41.3	37.9
PLP+TF(CEP+AUX-C)	Pitch	O	41.4	38.2
	Pitch	H	41.2	37.9
PLP+TF(CEP+AUX-A)	Energy	O	40.9	38.0
PLP+TF(CEP+AUX-C)	Energy	O	40.8	38.2
	Energy	H	40.9	38.0
PLP+TF(CEP+AUX-A)	ROS	O	41.3	37.9
PLP+TF(CEP+AUX-C)	ROS	O	42.0	38.3
	ROS	H	41.3	37.8

Table 5.6. Results of continuous speech recognition studies using concatenated PLP and Tandem features. The performance is measured in terms of word error rate (expressed in %). AF denotes auxiliary features, O denotes the auxiliary feature is observed and H denotes the auxiliary feature is hidden. Pitch - Pitch frequency, Energy - Short-term energy and ROS - Rate-of-speech. Bold face indicates the best system.

of auxiliary feature. However, if the acoustic model is already robust to certain variabilities which can be reduced by the integration of an auxiliary feature, then conditioning the acoustic model by that auxiliary feature may not help.

In order to verify this, we did experimental studies with TANDEM system which has been shown to be robust to speaker variability (Zhu *et al.*, 2004). The subword units used in this study are context-independent phonemes[4].

We used the "Tandem(CEP)+AUX" approach to integrate auxiliary feature for our studies. In this approach, the tandem feature sequence Z is extracted from the ANN trained only with standard feature and the tandem feature is modelled along with auxiliary feature in HMM/DBN-GMM system. Refer to Section 4.5.3 for more details. We trained different TANDEM systems corresponding to different auxiliary features pitch frequency, short-energy and ROS for Numbers task. The different systems are:

- Baseline (57 k parameters): The standard HMM/DBN-GMM system using only tandem feature vector (z_n).
- System $z_n \perp\!\!\!\perp a_n \,|\, q_n$ (58 k parameters): HMM/DBN-GMM system integrating auxiliary feature a_n where, a_n is treated as an additional feature (i.e., concatenated to z_n).
- System $a_n \perp\!\!\!\perp q_n \,|\, z_n$ (89 k parameters): HMM/DBN-GMM system integrating auxiliary feature a_n where, a_n conditions the emission distribution.

The results of this study for different auxiliary features pitch frequency, short-term energy and ROS are given in Table 5.7. During recognition, the auxiliary feature was hidden according to (4.21).

We observe from the results that for all the systems where the auxiliary features condition the emission distribution perform worse, where as, concatenation of the auxiliary features with tandem features yield significant improvement. This suggests that it is better to condition the acoustic models on any auxiliary feature only when the auxiliary feature is able to explain the variabilities present in the speech signal that the original acoustic model can not handle, otherwise concatenating them to standard features may help in improving the performance of ASR.

Moreover, we observe that the TANDEM baseline system with context-independent phoneme as subword units yield performance similar to the HMM/DBN-GMM PLP baseline system with context-dependent subword units and, there is further improvement in the performance of the system by using auxiliary feature as additional feature.

[4]The earlier TANDEM system studies presented in 5.2.3 were done with context-dependent phoneme units.

Systems	O or H	Pitch	Energy	ROS
Baseline	-	7.4		
$z_n \perp\!\!\!\perp a_n \mid q_n$	O	**6.4**	6.3	**6.1**
$z_n \perp\!\!\!\perp a_n \mid q_n$	H	6.7	**6.1**	6.2
$a_n \perp\!\!\!\perp q_n \mid z_n$	O	7.9	8.2	7.6
$a_n \perp\!\!\!\perp q_n \mid z_n$	H	8.0	7.9	7.9

Table 5.7. (Word error rate expressed in percentage for the Baseline HMM/DBN-GMM system and for the two HMM/DBN-GMM systems integrating auxiliary features on Numbers task. The subword units are context-independent phonemes. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (a) was used for Baseline. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (b) was used for $z_n \perp\!\!\!\perp a_n \mid q_n$. The HMM/DBN-GMM system corresponding to the BN in Figure 4.2 (d) for $a_n \perp\!\!\!\perp q_n \mid z_n$. Notations: z_n = tandem features, a_n = auxiliary feature (pitch frequency, short-term energy and ROS), O means a_n is observed and H means a_n is hidden. Bold face indicates the best system for each auxiliary auxiliary.

5.5 Summary and Conclusion

In this chapter, we studied context-dependent phoneme-based ASR systems integrating auxiliary features on connected word recognition task and continuous speech recognition task. Studies conducted in the framework of hybrid HMM/ANN systems, HMM/DBN-GMM systems and TANDEM systems show that:

- Conditioning the emission distribution upon auxiliary features pitch frequency, short-term energy and ROS does not help in improving the performance of ASR when context-dependent phonemes are used as subword units.
- Using the auxiliary features as additional features can help in improving the performance of ASR.
- TANDEM system performs better than the conventional PLP-based HMM/GMM system in both clean and noisy conditions. The performance can be improved further by integrating auxiliary features in TANDEM system.
- In TANDEM framework, it is better to jointly model the tandem-feature and auxiliary feature (“Tandem(CEP)+AUX” approach) rather than integrating the auxiliary feature through hybrid HMM/ANN system and then, extracting the tandem-features (“Tandem(CEP+AUX)” approach).
- Though integrating auxiliary features in standard ASR system yielded improvement in the performance for Numbers Task, it did not yield similar improvements for CTS task.

Chapter 6

Pronunciation Model Evaluation

6.1 Introduction

In Chapters 4 and 5, we have shown how to improve the acoustic model by integrating auxiliary features. In the present chapter, we introduce a way to evaluate the quality of baseform pronunciation models. Through the proposed approach, we show that by integrating auxiliary features in standard ASR the matching properties of the acoustic observation sequences and the pronunciation model of the words can be improved. As a by product, this approach allows to extract new pronunciation variants for the words, which when added to the lexicon improves the performance of the ASR significantly.

This chapter is organized as follows. In Section 6.2, we give an overview of pronunciation variation modelling. We introduce and motivate the proposed approach to evaluate the adequacy of the baseform pronunciation model in Section 6.3. The proposed approach involves extraction of new pronunciation variants and evaluating them through some evaluation measures. Section 6.4 describes the HMM inference approach to extract new pronunciation pronunciation variants. Section 6.5 describes the measures used to evaluate the pronunciation variants. Sections 6.6 and 6.7 present the experimental setup and analytical studies, respectively. Section 6.8 presents the ASR studies with pronunciation variants extracted via the proposed approach. In Section 6.9, we summarize and conclude.

6.2 Pronunciation Modelling

The lexicon of an ASR system contains the words and their standard pronunciations, i.e. a sequence of subword units (Ostendorf, 1999), usually phonemes. We refer to this sequence of subword units of a word as the baseform pronunciation of the word. The baseform pronunciation of each word is generally obtained from a standard lexical dictionary which contains both the meaning of the word and the way the word is to be pronounced. The baseform pronunciation model could be further enriched by phonological rules.

In standard HMM-based ASR, decoding involves stochastic pattern matching between the acoustic observation sequence X and pronunciation model (sequence of subword units) given the acoustic models. In phoneme-based ASR systems, it is generally expected that speaker pronounce the words according to the phonetic transcription given in the lexicon. However, during conversation speaker do introduce pronunciation variation, which leads to a mismatch between the acoustic observation and pronunciation model. Pronunciation variations can occur at different levels (Strik and Cucchiarini, 1999):

1. The acoustic characteristic level due to speaking style, speaking rate, different accent, pitch, differences in the length of the vocal tract, background noise (Lombard effect), emotion or

stress.

2. The lexical characteristic level due to phonological processes such as assimilation, co-articulation, reduction, deletion and insertion, accent or "*liaisons*" in French.

For these reasons, single baseform pronunciation is not sufficient to handle pronunciation variation. Sometimes, even with high frame/phoneme level performance the word performance can still be poor because the lexical constraints in the baseform pronunciation model are not correct.

There are different ways to improve the match between acoustic parameters and pronunciation models, such as:

1. Adapting or enriching the pronunciation models. For instance, generating new pronunciation variants and adding them to the lexicon or creating pronunciation lattices (Strik and Cucchiarini, 1999). In the literature, good performance has been reported with limited number of pronunciation variants, often less than 1.1 pronunciations per word (Strik and Cucchiarini, 1999; Kessens *et al.*, 1999, 2003).

2. Adapting the acoustic model, such as iterative training (Strik and Cucchiarini, 1999), sharing the parameters of the phoneme models in baseform pronunciation with parameters of the phonemes in alternate realization(s) (Sarclar, 2000).

3. Extracting subword units and word pronunciations automatically from the data (Bacchiani and Ostendorf, 1999; Singh *et al.*, 2002).

The most common practice is to generate new pronunciation variants. The approaches used for generating new pronunciation variants can be broadly classified as, (a) knowledge-based (b) data-driven approaches, or (c) a mix of both (Strik and Cucchiarini, 1999). The generated pronunciations are kept separate (Strik and Cucchiarini, 1999) or merged into a single (more complex) HMM (Stolcke and Omohundro, 1994). These pronunciation variants can also be pruned/smoothed to keep only the most representative ones. However, while this improves the matching properties of each of the words individually, the way these multiple pronunciations are defined is also known to increase the confusion between words.

6.3 Proposed Approach

In this chapter, we propose an alternative approach where (Magimai-Doss and Bourlard, 2001; Magimai.-Doss and Bourlard, 2005):

- We evaluate the adequacy of the baseform pronunciation of words in the lexicon. In other words, how well the baseform pronunciation matches the acoustic models given the acoustic observation.

- When the baseform pronunciation is inadequate for a given word, pronunciation variants that are as stable as possible to acoustic variability, and at the same time not too dissimilar to the baseform pronunciation, are extracted and added to the lexicon.

The adequacy of a given baseform pronunciation model is evaluated by (1) relaxing the lexical constraints of the baseform pronunciation, (2) inferring new pronunciation variants from the relaxed HMM and (3) measuring the confidence level of the acoustic match for the inferred pronunciation models. The proposed approach can be summarized as follows:

1. Relaxing lexical constraint: An ergodic HMM[1] is initialized to only allow the generation of a first order approximation of the baseform pronunciation. The constrained ergodic HMM is

[1] An ergodic HMM contains a set of fully-connected phonetic states with arbitrary transition probability matrix, where, every state can be reached from every other state in a finite but aperiodic number of steps (Rabiner and Juang, 1993).

later relaxed iteratively to converge towards an ergodic HMM with uniform transition probabilities.

2. Inference of pronunciation variants: For each of the (relaxed) ergodic HMM configurations, a phonetic transcription is generated (pronunciation variant) by HMM inferencing and evaluated.

3. Evaluation: The inferred pronunciation variants are evaluated by,

   - Confidence: estimating the confidence level of the inferred pronunciation variant, i.e., how well the acoustic observation matches the pronunciation model.
   - Levenshtein distance: Computing Levenshtein distance between the inferred phonetic sequence and the associated baseform pronunciation.

   Here, we basically assess the "stability" of the baseform pronunciation to perturbations through the confidence measure and Levenshtein distance obtained. In other words, how fast the inferred pronunciation variant diverges from the baseform pronunciation as the ergodic HMM is relaxed (slow the divergence more stable is the pronunciation model).

6.4 Relaxing and Inferring Pronunciation Models

HMM inference is a technique to infer the "best" HMM model associated with a given set of utterances (Mokbel and Jouvet, 1998). This inference is generally done by performing unconstrained subword-unit level decoding of the utterance, matching the acoustic sequence X on an *ergodic* HMM model with uniform transition probabilities. Here an ergodic HMM model with uniform transition probabilities will be referred to as fully ergodic HMM. For our studies, we used hybrid HMM/ANN system and each HMM state q_n corresponds to a context-independent phoneme which is associated with a particular ANN output. Figure 1 shows a 3-state ergodic HMM model, including the non-emitting initial and final states I and F.

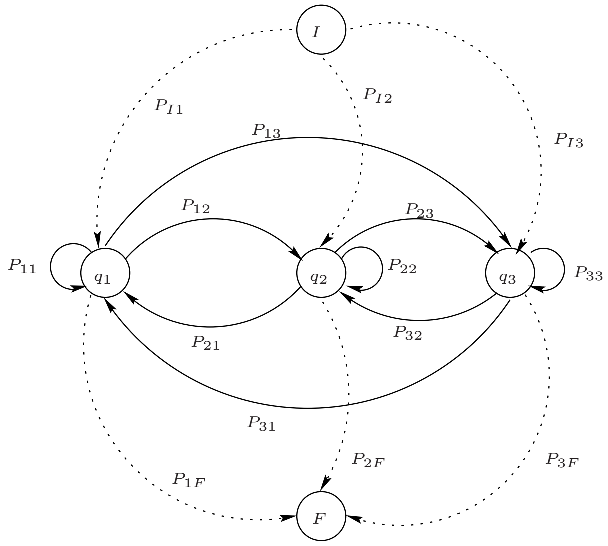


Figure 6.1. 3-state Ergodic HMM

A fully ergodic HMM is capable of producing any state sequence (since there is no grammar or lexical constraint in it), as opposed to a left-to-right HMM which can only produce constrained state sequences. A fully ergodic HMM is obviously too general to model lexical constraints. In current

ASR systems, the words are usually represented as left-to-right sequences of subword-units. For example, Figure 6.2 illustrates a word represented by pronunciation $\{q_2, q_1, q_2\}$.

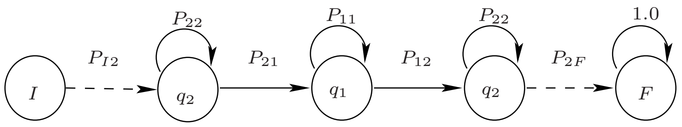


Figure 6.2. Left-to-Right HMM

The transition probability matrix for the fully ergodic HMM in Figure 6.1 is

$$T = \begin{bmatrix} P_{II} & P_{I1} & P_{I2} & P_{I3} & P_{IF} \\ P_{1I} & P_{11} & P_{12} & P_{13} & P_{1F} \\ P_{2I} & P_{21} & P_{22} & P_{23} & P_{2F} \\ P_{3I} & P_{31} & P_{32} & P_{33} & P_{3F} \\ P_{FI} & P_{F1} & P_{F2} & P_{F3} & P_{FF} \end{bmatrix} = \begin{bmatrix} 0.00 & 0.33 & 0.33 & 0.33 & 0.00 \\ 0.00 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \tag{6.1}$$

In pronunciation modelling literature, the HMM inference approach is used to generate pronunciation variants (Mokbel and Jouvet, 1998), by performing phonetic decoding (inference) of several utterances of the same/different words through the fully ergodic HMM which does not encode any lexical constraints.

On the contrary, the proposed approach to evaluate pronunciation model is based on a HMM inference mechanism which uses the prior knowledge of baseform pronunciation. For each lexicon word, and starting from its baseform pronunciation, we perform the following steps:

1. We first start from a transition matrix representing a first-order approximation of the baseform pronunciation (thus only allowing the transitions present in the left-to-right HMM). This is done by taking the transition probability of a fully ergodic HMM, say (6.1), adding an ϵ to the transitions present in the baseform pronunciation (e.g., Figure 6.2) followed by a renormalization, and thus yielding a transition matrix such as in (6.2). This ergodic model is referred to here as a *constrained ergodic model*.

$$T = \begin{bmatrix} 0.0 & \frac{1}{3+3\epsilon} & \frac{1+3\epsilon}{3+3\epsilon} & \frac{1}{3+3\epsilon} & 0.0 \\ 0.0 & \frac{1}{4+4\epsilon} & \frac{1+4\epsilon}{4+4\epsilon} & \frac{1}{4+4\epsilon} & \frac{1}{4+4\epsilon} \\ 0.0 & \frac{1+4\epsilon}{4+8\epsilon} & \frac{1}{4+8\epsilon} & \frac{1}{4+8\epsilon} & \frac{1+4\epsilon}{4+8\epsilon} \\ 0.0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{6.2}$$

It is easy to see that for a large value of ϵ, this *constrained (left-to-right) ergodic model* is a first order approximation of the baseform pronunciation.

2. Starting from a large value of ϵ, the *constrained ergodic model* is then slowly relaxed by decreasing the value of ϵ. For $\epsilon = 0.0$, this model is then equivalent to a fully ergodic HMM.

We note here that when a constrained ergodic HMM is used for inference, it can still recognize state sequences other than the baseform pronunciation because of its first order Markov assumption. For example, in the above example of a constrained ergodic HMM, the sequences allowed by the baseform pronunciation are:

$$\begin{aligned} I & \; -> \; q_2 \\ q_2 & \; -> \; q_1 \\ q_1 & \; -> \; q_2 \\ q_2 & \; -> \; F \end{aligned}$$

Hence, the constrained ergodic HMM can also recognize state sequences such as $\{q_2, q_1, q_2, q_1, q_2\}$ or just q_2, apart from the intended state sequence $\{q_2, q_1, q_2\}$.

A constrained ergodic HMM encodes the lexical constraint information through the transitional probability matrix. When the ϵ value is decreased the lexical constraint is relaxed such that the transition probability matrix starts allowing transitions which are not present in the baseform pronunciation. It is easy to see that the fully ergodic HMM is a special case of constrained ergodic HMM which does not have any lexical constraint information.

The underlying idea exploited in the present work thus, consisted in generating for each utterance of a given lexicon word several pronunciation variants through successive relaxation of the transition matrix, i.e., decreasing the value of ϵ. The quality of these inferred pronunciation variants are then assessed against the observed data in terms of different measures. In the following section, we describe the different evaluation measures that are used to assess the quality of the pronunciation variants.

6.5 Evaluation Measures

In the previous section, we described how the ergodic HMM is relaxed and new pronunciation variants are inferred. In the proposed approach, the new pronunciation variants are evaluated based upon:

- Reliability: The reliability of the pronunciation variant is measure by an acoustic confidence measure.
- Proximity: The proximity is measured by computing the Levenshtein distance between the baseform pronunciation and the inferred pronunciation variant.
- Combination of the confidence measure and Levenshtein distance.

6.5.1 Confidence Measure

In the literature, different confidence measures that can be derived from a hybrid HMM/ANN system based on local phone posterior probabilities, $P(q_n = k|x_n)$ have been suggested (Williams and Renals, 1999; Magimai-Doss and Bourlard, 2001), where x_n is the feature vector at time frame n and $q_n = k$ is the state hypothesis. We use the posterior probability based confidence measure. The posterior based confidence measure is defined as the normalized logarithm of the segment-based accumulated posterior probabilities.

For a given segmentation (resulting in our case from a Viterbi algorithm using local posterior probabilities), we define the accumulated posteriors for all the acoustic vectors observed on state $q_n = k$ as:

$$CM_{post}(q_k) = \prod_{n=b_k}^{n=e_k} P(q_n = k|x_n), \tag{6.3}$$

where b_k and e_k are the begin and end frames of a state hypothesis $q_n = k$. Defining minus log of $CM_{post}(q_n = k)$ as the state-based confidence measure, we obtain:

$$\mathcal{CM}_{post}(q_n = k) = \sum_{n=b_k}^{n=e_k} \log P(q_n = k|x_n) \tag{6.4}$$

The probability of a decoding hypothesis is always underestimated due to the observation independence assumption. This underestimate creates a bias towards shorter decoding hypotheses. Duration normalization counteracts this bias. The duration normalized word-level posterior probability based confidence measure is then defined as:

$$\mathcal{CM}_{wpost} = \frac{1}{K} \sum_{k=1}^{k=K} \frac{\mathcal{CM}_{post}(q_n = k)}{e_k - b_k + 1}, \tag{6.5}$$

where, K is the number of constituent phonemes in the inferred model. The average posterior probability avg_p can then be computed from $\mathcal{CM}_{wpost}$ as

$$avg_p = e^{(\mathcal{CM}_{wpost})} \tag{6.6}$$

6.5.2 Levenshtein Distance

In the proposed approach, apart from measuring the confidence level, the proximity between the inferred pronunciation variant and the baseform pronunciation is of equal interest. We measure the proximity between the inferred pronunciation variant and the baseform pronunciation in terms of Levenshtein distance (LD).

Given two strings, the Levenshtein distance is defined as the minimum number of changes (substitutions, insertions and deletions) that has to be made in one string to convert it into another string (Sankoff and Kruskal, 1999). Consider two strings $/c/$ $/a/$ $/t/$ and $/a/$ $/c/$ $/t/$, in this case the Levenshtein score is two as a minimum of two changes have to be made to convert any one of the strings into another.

6.5.3 Combined Measure

The proposed approach evaluates the inferred pronunciation model based on both reliability and proximity, as, it is possible to infer a pronunciation variant with high confidence level that is completely different from the baseform pronunciation. Hence, we need to define a combined measure $comb$ which can be computed at each relaxation and inference step (i.e. for each value of ϵ). Motivated by the approach proposed in (Beyerlein, 1998) for discriminative model combination, we compute $comb$ in the following manner:

$$comb = -\mathcal{CM}_{wpost} + \log(1 + LD) \tag{6.7}$$

Taking $\log(1 + LD)$ is appropriate as LD is an integer and has a wide dynamic range compared to $\mathcal{CM}_{wpost}$. Also, we are interested in changes in LD at lower levels (i.e. the deviations that are not too far from the baseform pronunciation) which the $\log$ function represents well. A high $comb$ value means low confidence i.e, $\mathcal{CM}_{wpost}$ and/or LD are high.

The proposed approach to evaluate baseform pronunciation model is used to:

1. Compare the quality of different acoustic models. In the previous two chapters we studied how to improve the acoustic model by integrating auxiliary features. Through the proposed approach, we show in Section 6.7 that integration of auxiliary feature in standard ASR improves the stability of the baseform pronunciation.

2. Extract new pronunciation variants which are both reliable (confidence level is high) and "close" enough to the baseform pronunciation (Levenshtein distance is low). We present ASR studies with the pronunciation variants in Section 6.8.

6.6 Experimental Setup

We use the PhoneBook speech corpus which contains isolated words spoken on average by 11-13 speakers for our studies (Pitrelli *et al.*, 1995). There are 42 context-independent phonemes including silence, each modelled by a single emitting state. The standard acoustic vector x_n is the 21 dimensional MFCCs extracted from the speech signal using an analysis window of 25 ms with a shift of 8.3 ms. For further details refer to Section 3.8.1. The auxiliary features used in this study are pitch frequency and short-term energy.

We use the following previously trained systems (Section 4.8.1) for our studies:

1. Hybrid HMM/ANN baseline system trained with standard features (*system-base*). This system corresponds to the system "Baseline" in Table 4.2.

2. Hybrid HMM/ANN systems trained with standard features and auxiliary features. In the previous chapter, we observed that these systems improve the performance of ASR systems. The auxiliary features are used in two different ways:

   (a) Concatenated to the standard feature to get an augmented feature vector with which hybrid HMM/ANN system is trained. The system trained with pitch frequency as auxiliary feature is denoted as *system-app-p* (System "x_n, a_n (no assumption)" in Table 4.2 for auxiliary feature pitch frequency), and the system trained with short-term energy as auxiliary feature is denoted *system-app-e* (System "x_n, a_n (no assumption)" in Table 4.2 for auxiliary feature short-term energy).

   (b) Auxiliary features conditioning the emission distribution. The system trained with pitch frequency as the auxiliary feature is denoted as *system-cond-p* (System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" in Table 4.2 for auxiliary feature pitch frequency), and the system trained with short-term energy as the auxiliary feature is denoted *system-cond-e* (System "$a_n \perp\!\!\!\perp q_n \,|\, x_n$" in Table 4.2 for auxiliary feature short-term energy).

These systems were trained for isolated word recognition. The words and speakers present in the training set, validation set and test set do not overlap. *system-app-p* and *system-cond-e* perform significantly better than *system-base*.

The following section presents the analytical studies which uses the evaluation measures described in the previous section to evaluate the baseform pronunciation model and use the proposed approach to compare the quality of different acoustic models.

6.7 Analytical Studies

We performed analytical studies using the acoustic models of *system-base*, *system-app-p* and *system-cond-e*. We used a part of the validation set, 75 words, each spoken on average by 12 different speakers. We performed evaluation of baseform pronunciations in the following manner:

1. For a given word utterance X, and given its known baseform pronunciation, initialize the $K \times K$ transition probability matrix (where $K = 44$ in our case, corresponding to the 42 phonemes, plus initial and final states) with a very large ϵ value to constrain the ergodic model to be equivalent to a first-order approximation of the baseform pronunciation of the word.

2. Perform forced Viterbi decoding based on that model using local posterior probabilities $P(q_k|x_n)$.

3. From the resulting best path, extract the phonetic level decoding.

4. Compute the average posterior avg_p using the best path as described earlier in Section 6.5.1.

5. Compute Levenshtein distance LD between the phonetic sequence obtained from step 3 and the baseform pronunciation.

6. Relax the underlying model towards a fully ergodic model ($\epsilon = 0$) by decreasing the ϵ value, and repeat steps 2-4 to infer new phonetic transcription and compute their associated avg_p and LD.

The ideal case suitable for ASR would be something like shown in Figure 6.3, where

1. When the inference is performed on a constrained ergodic HMM, the Levenshtein distance is zero and confidence level is good, typically $avg_p > 0.5$.

2. As the constrained ergodic HMM is relaxed to fully ergodic HMM, the inferred pronunciations diverge less from the baseform pronunciation.

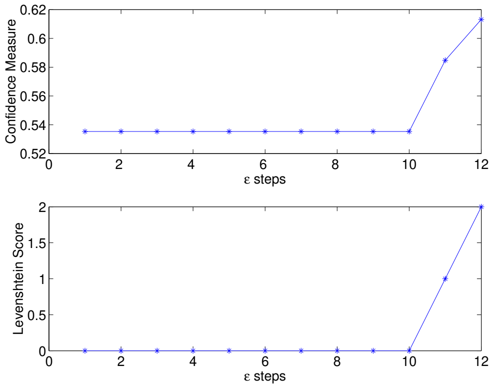


Figure 6.3. A case where the baseform pronunciation of word keeble in PhoneBook database uttered by a female speaker matches well with the acoustic observation. The inference was done with acoustic models of *system-app-p*.

But, in practice we observe the following (see Figure 6.4):

1. When the inference is performed on a constrained ergodic HMM (i.e. large value for ϵ) the confidence level is low and the Levenshtein distance is low.

2. As the constrained ergodic HMM is relaxed to fully ergodic HMM, the confidence level increases and the Levenshtein distance also increases rapidly, i.e., the inferred pronunciation variant quickly diverges from the baseform pronunciation.

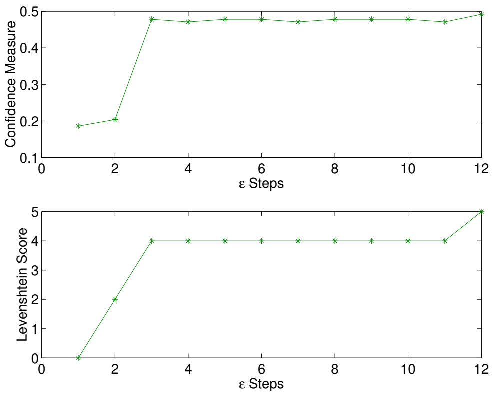


Figure 6.4. A case where the baseform pronunciation of word keeble in PhoneBook database uttered by a female speaker doesnot match well with the acoustic observation. The inference was done with acoustic models of *system-app-p*.

As can be observed from the Figures 6.3 and 6.4 with posterior-based confidence and Levenshtein distance together we can analyze the stability of the baseform pronunciation.

We evaluated the baseform pronunciation of all the 75 words using their multiple utterances with the procedure described earlier in this section. We did the same evaluation with acoustic models of *system-base*, *system-app-p* and *system-cond-e*. The main outcomes of this analytical study are the following:

1. When the baseform pronunciation of a word matches acoustic observations well, the evaluation across different speakers mostly yields a behavior similar to Figure 6.3, i.e., confidence level is high and, when relaxing the lexical constraints, the speed of divergence is slow.

2. When a baseform pronunciation is inadequate, the confidence level is low and the speed of divergence is fast for most of the utterances of the word.

3. When comparing across the acoustic models *system-base*, *system-app-p* and *system-cond-e*, most of the times the acoustic models trained with both standard features as well as auxiliary features (*system-app-p* and *system-cond-e*) match the baseform pronunciation well compared to acoustic models trained only with standard features (*system-base*). The comparison was performed using *comb*. The result of the comparison is illustrated in Figures 6.5 and 6.6.

Figures 6.5 and 6.6 show that integrating auxiliary feature in standard ASR improves the "stability" of the baseform pronunciation model. In other words, as the lexical constraints of the baseform pronunciation model are relaxed, for systems integrating auxiliary features (*system-app-p* and *system-cond-e*), the inferred pronunciation variant diverges from the baseform pronunciation model slowly as compared to the systems trained with only standard acoustic features (*system-cond-e*).

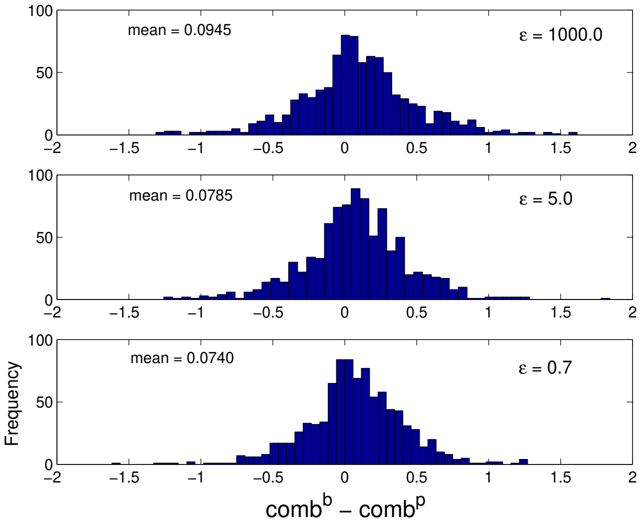


Figure 6.5. Histogram of difference between the $comb^b$ value (obtained by using acoustic models of *system-base*) and $comb^p$ value (obtained by using acoustic models of *system-app-p*) for different values of ϵ, for all the utterances. The means of $comb^b$ and $comb^p$ are statistically different (t-test, 5% confidence interval).

6.8 Pronunciation Variants Extraction

In this section, we describe experimental studies using pronunciation variants which are extracted using the proposed approach. The pronunciation variants are extracted by:

- Using the proposed approach to evaluate the adequacy of the baseform pronunciation models.

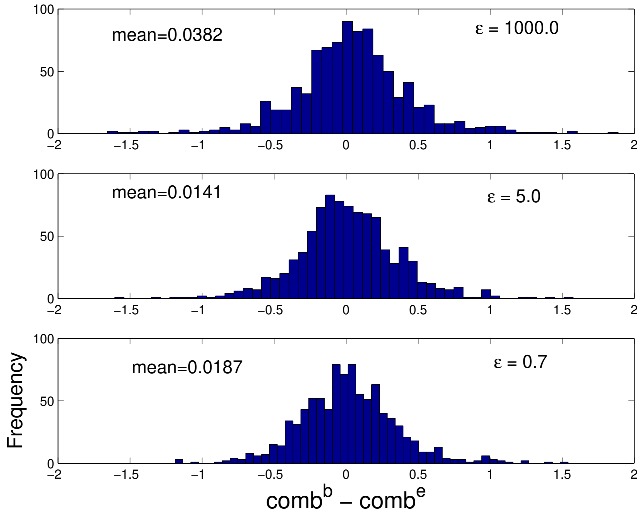


Figure 6.6. Histogram of difference between the $comb^b$ value (obtained by using acoustic models of *system-base*) and $comb^e$ value (obtained by using acoustic models of *system-cond-e*) for different values of ϵ, for all the utterances. The means of $comb^b$ and $comb^e$ are not statistically significant (t-test, 5% confidence interval).

- Extracting pronunciation variants that are reliable (high confidence level) and close enough to the baseform pronunciation (low Levenshtein distance)

In order to extract pronunciation variants for the words in the test set, we split the test set randomly (keeping the gender balance) into two parts:

1. "*H-set*": This set is used for baseform pronunciation evaluation and pronunciation variant extraction (45% of the original test set).

2. "*T-set*": This set is used for recognition studies (55% of the original test set). Since each speaker has spoken each word only once, the speakers present in the *H-set* of any word are not present in the *T-set* of that word.

The recognition performance of different systems on *T-set* for 8 different sets of 75 words lexicon and one set of 602 words lexicon are given in Table 6.1. The performance is measured as the average word error rate over the 8 lists in the test sets. In the previous section, we observed that integrating auxiliary feature in standard ASR system improves the matching and discriminating propoerties of the baseform pronunciation model. This also gets reflected in ASR performance, as it can be seen in the Table, *system-app-p* and *system-cond-e* perform better than the system *system-base*.

We extracted the pronunciation variants using the acoustic models of *system-app-p*, as this system performs better than all the systems and, also it better matches the acoustic observation and baseform pronunciation (as observed in last section).

6.8.1 Manual Pronunciation Variants Extraction

For the utterances of each word in *H-set*, we ran the following procedure to extract pronunciation manually:

1. If it is found that for the majority ($\geq 50\%$) of utterances of the word the baseform pronunciation is adequate (i.e., $avg_p \geq 0.5$ and $LD \leq 1$). Then, no pronunciation variants are included.

2. If the above condition is not satisfied, we look for the most frequently inferred pronunciation variant (not diverging far from the baseform, $LD \leq 2$) across different utterances and, add it

Systems		Performance 75 words	Performance 602 words
system-base		4.2	11.0
system-app-p	O	**2.5**	**7.3**
system-cond-p	O	3.5	9.9
	H	4.0	11.3
system-app-e	O	5.3	13.3
system-cond-e	O	2.9	8.3
	H	3.5	10.2

Table 6.1. Recognition studies performed on 8 different sets of 75 words lexicon and one set of 602 words lexicon with single pronunciation for each word. Performance is measured in terms of word error rate (WER), expressed in %. Notations: O: Auxiliary feature observed, H: Auxiliary feature hidden (i.e. integrated over all possible values of auxiliary feature).

to the lexicon. If there is no commonly inferred pronunciation (it most often happens for short words.), we extract variants from each utterance such that the confidence level is high (avg_p close to 0.5 or above) and at the same time LD is low (≤ 2). In other words, we only keep the pronunciation variants which are reliable and close to the baseform pronunciation.

The statistics of the test lexicon after adding the pronunciation variants is given in Table 6.2.

# of resulting Pronunciation models	Number of words
1	441
2	106
3	48
4	7

Table 6.2. Statistics of test lexicon: The pronunciation selection was done manually. The first column mentions the number of pronunciations and the second column gives the number of words with that number of pronunciations.

We performed recognition studies with the updated lexicon(s). The results of the recognition studies performed on 8 different sets of 75 words lexicon and one set of 602 words lexicon are given in Table 6.3. Comparing the performance of the respective systems in Tables 6.1 and 6.3, we

Systems		Performance 75 words	Performance 602 words
system-base		$3.0^{\dagger}$	$10.1^{\dagger}$
system-app-p	O	$\mathbf{1.7}^{\dagger}$	$\mathbf{6.4}^{\dagger}$
system-cond-p	O	$2.8^{\dagger}$	$9.2^{\dagger}$
	H	3.3	$10.7^{\dagger}$
system-app-e	O	$4.3^{\dagger}$	$12.0^{\dagger}$
system-cond-e	O	$2.3^{\dagger}$	7.9
	H	$2.7^{\dagger}$	$9.2^{\dagger}$

Table 6.3. Recognition studies performed on 8 different sets of 75 words lexicon and one set of 602 words lexicon with multiple pronunciations. The pronunciation selection was done manually. Performance is measured in terms of WER (expressed in %). Notations: O: Auxiliary feature observed, H: Auxiliary feature hidden. † Improvement in the performance is significant compared to the results in Table 6.1 (with 95% confidence or above)

observe that by adding new pronunciation variants we improve the performance of the 75 words lexicon system significantly. Improvements are also obtained in the case of one set of 602 words lexicon. This indicates that the selection of pronunciation variants that are reliable and close to baseform pronunciation does not increases the confusion between the words.

6.8.2 Automatic Pronunciation Variants Extraction

We also performed recognition studies by automatically selecting new pronunciation variants. The automatic selection of new pronunciation variants was done in the following manner:

1. For each utterance of each word in *H-set*, run the baseform pronunciation evaluation procedure. Sort the $comb$ score (6.7) obtained in ascending order and, select the pronunciation variant with lowest $comb$ score and $LD > 0$. If the baseform pronunciation model is stable, it is very much possible that $LD = 0$ always and thus, no pronunciation variants are selected.

2. For each word, rank these selected pronunciation variants in ascending order and, select the top two. This results in utmost two possible pronunciation variants for each word.

3. The pronunciation variants selected in Step 2 are added to the lexicon if avg_p is greater than 0.5. If avg_p lies between 0.45 and 0.5 then the pronunciation variant is selected, if the frame average posterior probability (obtained by summing the posterior probabilities of the states in the best path and dividing the sum by number of frames) of the best path is greater than 0.5. Otherwise, the pronunciation variants are rejected. This way only pronunciation variants that are reliable and close to the baseform pronunciation are selected.

The statistics of the test lexicon after adding the automatically extracted pronunciation variants is given in Table 6.4. Compared to manual selection there are more pronunciation variants. This is mainly due to the condition $LD > 0$ as opposed to $LD > 1$ in case of manual selection.

# of resulting Pronunciation models	Number of words
1	183
2	292
3	127

Table 6.4. Statistics of test lexicon: The pronunciation selection was done automatically. The first column mentions the number of pronunciations and the second column gives the number of words with that number of pronunciations.

The recognition studies were performed on updated lexicon(s). The results are given in Table 6.5. We observe that the automatically selected pronunciation variants also leads to similar improvements as that of manually selected pronunciation variants.

Systems		Performance 75 words	Performance 602 words
system-base		$3.0^\dagger$	$10.3^\dagger$
system-app-p	O	$\mathbf{1.8}^\dagger$	$\mathbf{6.4}^\dagger$
system-cond-p	O	$2.7^\dagger$	$8.9^\dagger$
	H	3.3	$10.1^\dagger$
system-app-e	O	$4.4^\dagger$	$12.0^\dagger$
system-cond-e	O	$2.4^\dagger$	7.6
	H	3.0	$9.3^\dagger$

Table 6.5. Recognition studies performed on 8 different sets of 75 words lexicon and one set of 602 words lexicon with multiple pronunciations. The pronunciation variants selection was done automatically. Performance is measured in terms of WER (expressed in %). Notations: O: Auxiliary feature observed, H: Auxiliary feature hidden. † Improvement in the performance is significant compared to the results in Table 6.1 (with 95% confidence or above)

6.9 Summary and Conclusion

In this chapter, we proposed an approach based on HMM inference to evaluate the adequacy of pronunciation models by:

1. Relaxing the lexical constraints of the baseform pronunciation model.
2. Inferring a new pronunciation variants for each relaxation.
3. Measuring the "stability" of the pronunciation model through a combination of acoustic confidence level measure and Levenshtein distance.

The proposed approach was used to:

- Compare the quality of different acoustic models, namely, acoustic models trained with only standard features and acoustic models trained with both standard features and auxiliary features.
- Extract new pronunciation variants that are reliable (high confidence level) and are "close" enough to baseform pronunciation (low Levenshtein distance).

Experimental studies conducted on isolated word recognition task shows that:

- Integrating auxiliary features in standard ASR improves the "stability" of the baseform pronunciation model, i.e., the matching and discriminating properties of the single baseform pronunciation model is improved.
- The ASR performance can be significantly improved by incorporating the selected pronunciation variants.

In this work, we have studied the proposed approach for pronunciation variant selection for isolated word recognition task which only contains with-in word pronunciation variation. Given the good results achieved on isolated work recognition task, we believe in future it would be interesting to further study the proposed approach in the context of large vocabulary continuous speech recognition system where both with-in and cross-word pronunciation variation is present.

Chapter 7

Using Graphemes as Subword Units in ASR

7.1 Introduction

Grapheme is a written symbol that is used to represent words, e.g., example alphabets in English language. In this chapter, we study the use of graphemes as subword units for ASR, particularly for the English language where, there is weak correspondence between the written form and spoken form compared to other languages such as Finnish or Spanish.

In Chapter 4, we studied how to model the joint distribution over hidden state space Q, observed feature space X, and some auxiliary source of knowledge A to improve the ASR performance. In this case, the auxiliary knowledge sources were particular acoustic features, such as pitch frequency, short-term energy and rate-of-speech. In the present chapter, we extend this strategy to jointly model phonemes, graphemes and standard features, where graphemes are now treated as the auxiliary source of knowledge. We initially studied this system for context-independent graphemes. The results from this study motivated us to further look into using context-dependent graphemes.

In Section 7.2, we motivate the use of grapheme in state-of-the-art ASR systems. We present an overview of research in this direction and motivate joint phoneme-grapheme based ASR. Section 7.3 presents the modelling process in phoneme-grapheme based ASR and Section 7.4 presents the studies conducted on phoneme-grapheme system using context-independent phonemes and graphemes. Section 7.5 presents our studies using context-dependent graphemes. Finally, Section 7.6 summarizes and concludes with our findings.

7.2 Motivation

State-of-the-art HMM-based ASR models the joint likelihood $p(Q, X)$, the evolution of the hidden state space Q and the observed feature space X over time. The states represent the subword units which describe the word model. Standard ASR systems typically use phoneme as subword units. The states represent the subword units (typically, phonemes) which describe the word model. The feature vectors are typically derived from the smoothed spectral envelope of the speech signal. In Chapter 4, we studied how to model the evolution of auxiliary knowledge source $A = \{a_1, \cdots, a_n, \cdots a_N\}$ along with Q and X, i.e. model $p(Q, X, A)$ instead of $p(Q, X)$. The auxiliary knowledge source that was mainly investigated were auxiliary features pitch frequency, short-time energy and rate-of-speech. In this chapter, we extend this strategy of modelling auxiliary source of knowledge to model additional subword units. Here, these additional subword units will be referred to as as auxiliary subword units, and the auxiliary subword units that we investigated are graphemes.

In recent studies, good results have been reported using graphemes as subword units for languages such as German, Dutch and Swedish (Schukat-Talamazzini *et al.*, 1993; Kanthak and Ney, 2002; Killer *et al.*, 2003). There are certain advantages in using graphemes as subword units, such as:

- The definition of the lexicon is easy, i.e., the orthographic transcription of the word can be easily derived.
- The word model representation is unique, e.g., the word $ZERO$ can be pronounced as /z/ /ih/ /r/ /ow/ or /z/ /iy/ /r/ /ow/, but the grapheme-based representation remains as $[Z][E][R][O]$.
- Graphemes could complement the phonetic information.
- There is no need for phonetic transcription.

While there are certain advantages in using graphemes as subword units, there are certain drawbacks too, such as:

- There is no obvious relationship with acoustic features. In other words, the acoustic feature vectors derived from the smoothed spectral envelope of the speech signal typically depict the characteristics of phonemes.
- There is a weak correspondence between the graphemes and the phonemes in languages such as English (Sejnowski and Rosenberg, 1987). For instance, the grapheme $[E]$ in word $ZERO$ associates itself to phoneme /ih/, where as, in word $EIGHT$ it associates itself to phoneme /ey/.

Finnish ASR system is an ideal example for a grapheme based ASR system as mismatches between the written form and the spoken format of words are quite exceptional (Kurimo, 1997). Thus, in Finnish ASR system although the speech is modelled by phonemes, they are written down as graphemes. The mismatch errors and some unmodelled rare phonemes have been found to increase phoneme error rate. More recent works in Finnish ASR are looking into other subword units such as syllable, morphs (Siivola *et al.*, 2003).

As mentioned earlier, unlike Finnish for other languages such as English, German Dutch there is no direct correspondence between written form and spoken form. (Schukat-Talamazzini *et al.*, 1993) used "polygraph" as subword units for word modelling, which is essentially letters-in-context similar to polyphones (phonemic units allowing preceding and following context of arbitrary length). Experimental studies conducted on continuous speech and isolated word recognition tasks showed that good results (better than context-independent phone) could be obtained using "polygraph" as subword units.

In a more recent study, an approach of explicitly mapping orthographic transcription to a phonetic one was investigated in the context of speech recognition (Kanthak and Ney, 2002). In this approach, the orthographic transcription of the words are used to map them onto acoustic HMM state models using phonetically motivated decision tree questions, e.g., a grapheme is assigned to a phonetic question if the grapheme is part of the phoneme. The decision tree was generated manually as well as automatically (using log-likelihood gain and observation count). Recognition studies were performed on databases of three different languages (Dutch, German and English). For Dutch and German, where there is stronger association between phonemes and graphemes, this approach yielded performance comparable to their respective phoneme-based ASR system. For English though, where the grapheme to phoneme mapping is more complex, the performance of the system was fairly poor compared to purely phoneme-based ASR system.

(Killer *et al.*, 2003), have investigated a context-dependent grapheme based speech recognition, where the context is modelled through a decision tree based clustering procedure (Killer *et al.*, 2003). Experimental studies conducted on English, German and Spanish languages yielded competitive results compared to phoneme-based system for German and Spanish languages, but fairly poor performance for English language.

In this chapter, we propose a phoneme-grapheme based ASR system that, during training, jointly models the phoneme and grapheme subword units. During recognition, the decoding is done either using one or both the subword units (Magimai.-Doss *et al.*, 2003b, 2004a). Basically, this can be seen as a system where word models are described by two different complimentary subword units, i.e., the phonemes and the graphemes (as shown in Figure 7.1).

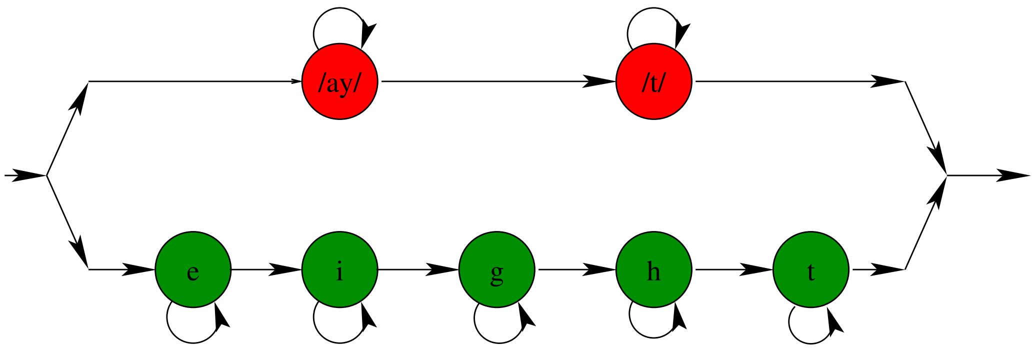


Figure 7.1. A word model in phoneme-grapheme based ASR. In standard ASR system, several states make up a phoneme. For simplicity in this figure we have represented every phoneme by a single state.

The architecture of the resulting system is then similar to factorial HMMs (Ghahramani and Jordan, 1997), where there are several chains of states as opposed to a single chain in standard HMMs. Each chain has its own states and dynamics; but the observation at any time depends upon the current state in all the chains (see Figure 7.2). (Logan and Moreno, 1997) were one of the first to use factorial HMM for ASR. They modified the factorial HMM where, the same discrete space was used for each chain and, each chain had different observations. This system did not yield promising results. In our case, instead of dividing states representing the same subword units into chains, there are two parallel chains, each corresponding to a specific subword unit representation and, the observation is same for both the chains. Similar models have also been used for ASR more recently, e.g. DBN-based multi stream speech recognition (Zhang *et al.*, 2003), modelling articulatory features (Wester *et al.*, 2004), multi-rate modeling of speech (Cetin and Ostendorf, 2005).

In the following section, we describe the modelling process of the phoneme-grapheme based ASR in detail.

7.3 Modelling in Phoneme-Grapheme Based ASR

Standard HMM-based ASR systems model the evolution of the observed acoustic feature sequence $X = \{x_1, \cdots, x_n, \cdots, x_N\}$ and the associated hidden state sequence $Q = \{q_1, \cdots, q_n, \cdots, q_N\}$ through the joint likelihood $p(Q, X)$[1] as

$$p(Q, X) \approx \prod_{n=1}^{N} p(x_n|q_n) \cdot P(q_n|q_{n-1}) \tag{7.1}$$

where $q_n \in \mathcal{Q}$, $\mathcal{Q} = \{1, \cdots, k, \cdots, K\}$. In the present work, q_n corresponds to the phoneme state sequence.

Similarly for a system with state sequence $L = \{l_1, \cdots l_n, \cdots l_N\}$ as the hidden space, we model

$$p(L, X) \approx \prod_{n=1}^{N} p(x_n|l_n) \cdot P(l_n|l_{n-1}) \tag{7.2}$$

[1]for all paths Q if path unknown

where $l_n \in \mathcal{L}$, $\mathcal{L} = \{1, \cdots, r, \cdots, R\}$. In the present work, l_n corresponds to the grapheme state sequence.

In phoneme-grapheme based ASR, we are interested in modelling the evolution of two hidden spaces Q and L (instead of just one) and the observed space X over time i.e., $p(Q, L, X)$. For such a system, the forward recurrence can be written as:

$$\begin{aligned}\alpha(n,k,r) &= p(q_n = k, l_n = r, X_1^n) \\ &= p(x_n|q_n = k, l_n = r) \sum_{i=1}^{K} P(q_n = k|q_{n-1} = i) \sum_{j=1}^{R} P(l_n = r|l_{n-1} = j)\alpha(n-1, i, j) \quad (7.3)\end{aligned}$$

where $\alpha(n, k, r)$ is the likelihood of being in phoneme state $q_n = k$ and grapheme state $l_n = r$ at time frame n having observed the acoustic sequence $X_1^n = \{x_1, \cdots x_n\}$ assuming conditional independence between Q and L given x_n. Usually in languages such as English, there is weak correspondence between phoneme state q_n and grapheme state l_n. In such cases, it may be possible to better model the relation between Q and L with another hidden variable such as word. Figure 7.2 illustrates with a graphical model representation of the approach. The acoustic observation x_n is conditioned upon the hidden states q_n and l_n. There are two chains, one corresponding to the phoneme state sequence Q and the other corresponding to the grapheme state sequence L.

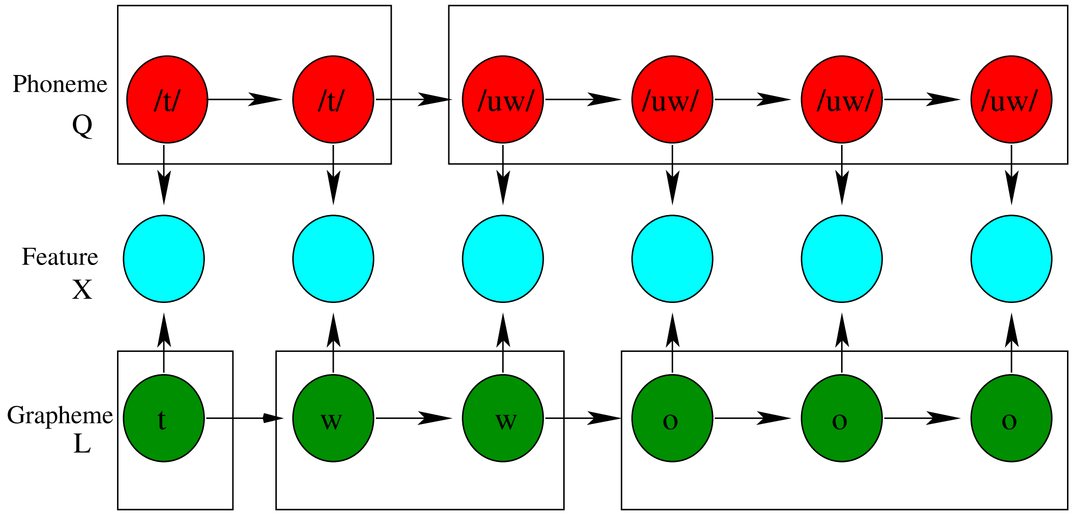


Figure 7.2. Graphical model representing acoustic modelling in phoneme-grapheme based ASR.

The likelihood of the data can then be estimated as:

$$p(X) = \sum_{k=1}^{K} \sum_{r=1}^{R} \alpha(N, k, r) \quad (7.4)$$

Finally, the Viterbi decoding algorithm that gives the best sequence in the Q and L spaces can be written as:

$$V(n,k,r) = p(x_n|q_n = k, l_n = r) \max_i P(q_n = k|q_{n-1} = i) \max_j P(l_n = r|l_{n-1} = j) V(n-1, i, j) \quad (7.5)$$

where $V(n, k, r)$ is the likelihood of the best path being in phoneme state $q_n = k$ and grapheme state $l_n = r$ at time frame n having observed acoustic sequence X_1^n.

The resulting decoding algorithm is thus equivalent to performing dynamic programming in a three dimensional space (phoneme, grapheme and time) unlike the conventional ASR where the

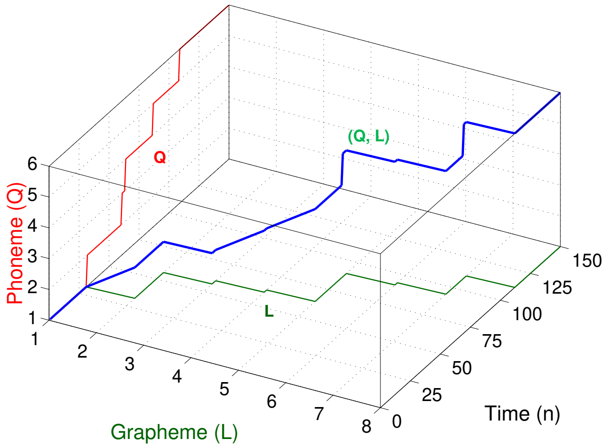


Figure 7.3. 3D Viterbi decoding in phoneme-grapheme based ASR.

dynamic programming is performed in two dimensional space (phoneme and time). Figure 7.3 illustrates the decoding process in phoneme-grapheme based ASR.

In state-of-the-art ASR, the emission distribution can be modelled by Gaussian mixture models (GMM) or artificial neural network (ANN). As described earlier in Chapter 3, in case of hybrid HMM/ANN ASR, a multilayer perceptron (MLP) is trained with K output units associated with $\mathcal{Q}$ for the system in (7.1). The likelihood estimate is replaced by the scaled-likelihood estimate which is computed from the output of the MLP (posterior estimates) and priors of the output units (hand counting). For instance, $p(x_n|q_n)$ in (7.1) is replaced by its scaled-likelihood estimate $p_{sl}(x_n|q_n)$, which is estimated as (Bourlard and Morgan, 1994):

$$p_{sl}(x_n|q_n) = \frac{p(x_n|q_n)}{p(x_n)} = \frac{P(q_n|x_n)}{P(q_n)} \tag{7.6}$$

We have investigated the proposed system in the framework of hybrid HMM/ANN ASR where, the emission distribution $p(x_n|q_n = k, l_n = r)$ could be estimated in different ways. A first solution consists in training an MLP with $K \times R$ output units[2] and estimate the scaled-likelihood as:

$$\frac{p(x_n|q_n = k, l_n = r)}{p(x_n)} = \frac{P(q_n = k, l_n = r|x_n)}{P(q_n = k, l_n = r)} \tag{7.7}$$

This scaled-likelihood estimate is then used to decode jointly in phoneme and grapheme space according to (7.5). During training, such a system would automatically model the association between the subword units in Q and L. This system has an added advantage that it can be reduced to a single hidden variable system by marginalizing any of the hidden variables, yielding

- The phoneme likelihood (by marginalizing out graphemes):

$$\frac{p(x_n|q_n = k)}{p(x_n)} = \frac{\sum_{j=1}^{R} P(q_n = k, l_n = j|x_n)}{P(q_n = k)} \tag{7.8}$$

- The grapheme likelihood (by marginalizing out phonemes):

$$\frac{p(x_n|l_n = r)}{p(x_n)} = \frac{\sum_{i=1}^{K} P(q_n = i, l_n = r|x_n)}{P(l_n = r)} \tag{7.9}$$

[2] It is equivalent to using unfactored $K \times R$ space

and using these scaled-likelihood estimates to decode according to (7.1) or (7.2), respectively.

The second solution consists in assuming independence between the two hidden variables,

- Training two separate MLPs. One corresponding to phoneme units with K output units and, the other corresponding to the grapheme units with R output units.
- Estimating the scaled-likelihood as following:

$$\begin{aligned}\frac{p(x_n|q_n = k, l_n = r)}{p(x_n)} &\approx \frac{P(q_n = k|x_n)P(l_n = r|x_n)}{P(q_n = k)P(l_n = r)} \\ &\approx p_{sl}(x_n|q_n = k)p_{sl}(x_n|l_n = r) \qquad (7.10)\end{aligned}$$

- The resulting scaled-likelihood estimate is then used too decode in the phoneme and grapheme space according to (7.5).

A comparison between performance of systems based on 7.7 and 7.10 for Numbers task can be found in Table 7.8.

7.4 Phoneme-Grapheme based ASR Studies

In this section, we report our experimental studies with Phoneme-Grapheme based ASR system on different ASR tasks, namely, isolated word recognition task (PhoneBook) and connected word recognition task (OGI Numbers95).

Though, the main focus of the present work is to jointly model phoneme units and grapheme units to improve the performance of the ASR system. Our focus is also upon improving the acoustic models of grapheme units by the integration of phonetic knowledge. So, we make use of the modelling approaches proposed in the last section to also study grapheme-based ASR. We train ASR systems that jointly model phoneme and grapheme information. During recognition, we marginalize out the phoneme information and use only grapheme information for recognition.

7.4.1 Isolated Word Recognition Task

We used PhoneBook database for task-independent speaker-independent isolated word recognition. The acoustic feature is 21 dimensional MFCC feature. Further details about the task and the definition of the training set, validation set and test set refer to Section 3.8.1 in Chapter 3.

There are 42 context-independent phonemes including silence associated with $\mathcal{Q}$, each modelled by a single emitting state. We trained a phoneme baseline (*System P*) system and performed recognition using single pronunciation of each word. The performance of the phoneme baseline system is given in Table 7.1.

There are 28 context-independent grapheme subword units associated with $\mathcal{L}$ representing the 26 characters in English, silence and + symbol present in the orthographic transcription of certain words in the lexicon where two words are joined. Similar to phonemes each of the grapheme units are modelled by a single emitting state. We trained a grapheme baseline system (*System G*) via embedded Viterbi training and performed recognition experiments using the orthographic transcription of the words. The performance of the grapheme baseline system is given in Table 7.1.

System	# of output units	WER
System P	42	4.7
System G	28	43.0

Table 7.1. Performance of phoneme and grapheme baseline systems on isolated word recognition task. The performance is measured in terms of word error rate (WER) expressed in %.

It can be observed from the results that the grapheme-based system performs significantly poorer as compared to the phoneme-based system. In (Kanthak and Ney, 2002), a similar trend was observed when comparing context-independent phoneme based ASR system and context-independent grapheme based ASR system. We performed grapheme-based ASR studies where, the phonetic knowledge is treated as auxiliary knowledge source to see if performance better than 43.0% can be achieved.

Grapheme-based ASR Studies

We first tried modelling the relation between the phoneme and grapheme automatically from the data by training a single MLP with $42 \times 28 = 1176$ output units. However, training such a large network is a difficult task. We thus trained ANNs with different configurations, all yielded poor results for both phoneme and grapheme. Hence, we took an alternate approach where the phoneme set is clustered into broad-phonetic-class representation. By broad-phonetic-class, we refer to the phonetic features, such as manner, place, height.

According to linguistic theory, each phoneme can be decomposed into some independent and distinctive features, and the combination of these features serves to uniquely identify each phoneme (Dalsgaard *et al.*, 1991; King and Taylor, 2000; Hosom, 2000). In our studies, we used the phonetic feature values similar to the one used in (Hosom, 2000, Chapter 7). Table 7.2 presents the different broad-phonetic-classes that we have used and their corresponding values. It can be seen from the table that the number of values for manner, place and height broad-phonetic classes are 10, 12, and 7, respectively. Thus, by collapsing the phonemes into a broad-phonetic-class, we can train a grapheme-broad-phonetic-class system capturing the relation between the graphemes and the values of the broad-phonetic-class. The mapping between the phonemes and the values of the broad-phonetic-classes was obtained from the *International Phonetic Alphabet (IPA) chart*[3] (IPA-Chart, 1996).

Broad-phonetic-class	Values
Manner	vowel, approximant, aspiration, nasal, stop, voiced stop, fricative, voiced fricative, closure, silence
Place	front, mid, back, retroflex, lateral, labial, dental, alveolar, dorsal, closure, unknown, silence
Height	maximum, very low height, low height, high height, very high height, closure, silence

Table 7.2. Different broad-phonetic classes and their values.

In our work, we studied three different grapheme-broad-phonetic-class systems corresponding to the different broad-phonetic classes:

1. *System GBM*: The phoneme units ($K = 42$) are mapped to the values of broad-phonetic-class manner ($K = 10$). A MLP with 10×28 output units is trained.

2. *System GBP*: The phoneme units ($K = 42$) are mapped to the values of broad-phonetic-class place ($K = 12$). A MLP with 12×28 output units is trained.

3. *System GBH*: The phoneme units ($K = 42$) are mapped to the values of broad-phonetic-class height ($K = 7$). A MLP with 7×28 output units is trained

The MLPs are trained via embedded Viterbi training and they have same number of parameters. During each training iteration, we marginalized out the broad-phonetic class as per (7.9) and performed Viterbi decoding according to (7.2) to get the segmentation in-terms of graphemes.

We then performed recognition studies just using graphemes as the subword units i.e., orthographic transcription of the words like the grapheme baseline system. This was achieved by:

[3]Thanks to Mark Barnard (IDIAP) for helping with his linguistic knowledge

1. Marginalizing out the broad-phonetic-class as per (7.9) to estimate the scaled-likelihoods of the grapheme units (i.e,. the broad-phonetic-class acts like an auxiliary knowledge source which is used during the training, but hidden during recognition.)

2. Performing decoding according to (7.2), like any standard ASR system.

The fourth column in Table 7.3 presents the experimental results of this study. The results show that the performance of grapheme-based ASR system can be significantly improved by the integration of phonetic knowledge, but still this performance is significantly poorer compared to phoneme-based ASR system.

System	Broad-phonetic-class	# of o/p units	WER
System G	-	28	43.0
System GBM	Manner	280	28.1
System GBP	Place	336	27.2
System GBH	Height	196	27.1

Table 7.3. Performance of grapheme-based ASR system using broad-phonetic-class as auxiliary source of knowledge on isolated word recognition task. The performance is measured in terms of word error rate (WER) and is expressed in %.

Phoneme-Grapheme ASR studies

Starting from the improved grapheme-based system, we then studied whether the grapheme information could help us to improve the performance of ASR if used as an auxiliary knowledge source. We investigated this in the lines of (7.10), where we assume independence between the phoneme units and grapheme units. We thus model the phoneme units and grapheme units by separate MLPs and, during recognition, multiply the scaled-likelihood estimates obtained from the two systems in order to estimate $p(x_n|q_n, l_n)$. We conducted recognition experiments by combining the scaled-likelihood estimates of the phoneme units and the scaled-likelihood estimates of the grapheme units estimated from different MLPs, corresponding to the grapheme baseline system and the different grapheme-broad-phonetic-class systems. This yielded slightly poorer performance compared to the phoneme baseline system.

It can be observed from (7.10) that the scaled-likelihood estimates of phoneme units and grapheme units can be interpreted as two different kinds of probability streams that are combined with equal weights. In literature, improved performance have been reported by weighting the log probability of multiple streams differently (Hagen, 2001; Misra *et al.*, 2003). The weights can be estimated automatically during recognition or can be a fixed weight.

In order to see how crucial the weights are in determining the performance of the system, we conducted an experiment where we fixed the weights and performed recognition experiments on the test set. Given the weight w, we estimated the scaled-likelihood in (7.10) as:

$$\frac{p(x_n|q_n = k, l_n = r)}{p(x_n)} \approx p_{sl}^{w}(x_n|q_n = k)p_{sl}^{w-1}(x_n|l_n = r) \tag{7.11}$$

We varied the weights in steps of 0.05 and performed recognition experiments at each step. The result of this study is shown in Figure 7.4. The best performance obtained was 4.1% for the case where the grapheme probabilities were estimated from the grapheme-broad-phonetic-class system using the place broad-phonetic-class as auxiliary source of knowledge. The resulting model is significantly better than the baseline system with 95% confidence. It can be seen from the figure that the operating points of the different systems are different. It is also closely related to how the grapheme-based systems perform individually.

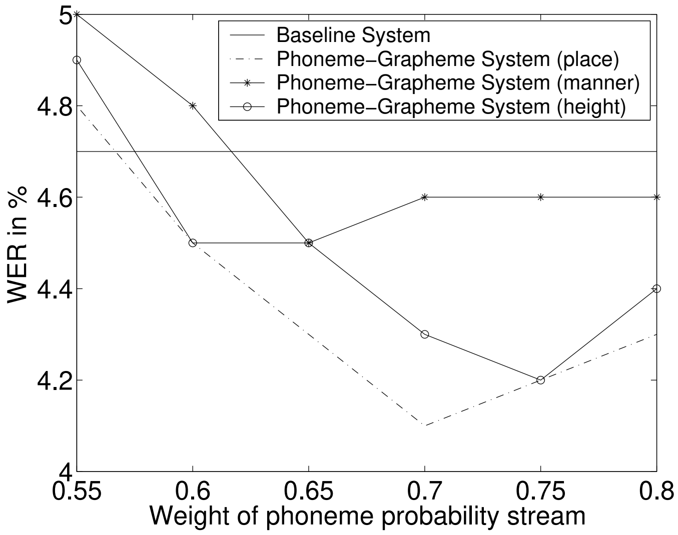


Figure 7.4. Plot illustrating the relationship between the weight and the word error rate (WER) of the phoneme-grapheme system on isolated word recognition task..

7.4.2 Numbers Task

We also used the OGI Numbers 95 database for phoneme-grapheme ASR studies. The acoustic vector x_n is the 39 dimensional PLP features, which were used for the previous experiments with auxiliary features presented in Chapters 4 and 5. For further details about the task including the definition of training set, validation set and test set refer to Section 3.8.2 in Chapter 3. All the MLPs trained have the same number of parameters.

There are 24 context-independent phonemes including silence associated with $\mathcal{Q}$, each modelled by a single emitting state. We trained a phoneme baseline system (*System P*) via embedded Viterbi training and performed recognition using single pronunciation of each word. The performance of the phoneme baseline system is given in Table 7.4.

There are 19 context-independent grapheme subword units including silence associated with $\mathcal{L}$ representing the characters in the orthographic transcription of the words. Similar to phonemes each of the grapheme units are modelled by a single emitting state. We trained a grapheme baseline system (*System G*) via embedded Viterbi training and performed recognition experiments using the orthographic transcription of the words. The performance of the grapheme baseline system is given in Table 7.4. Again, the phoneme baseline system performed significantly better than the grapheme baseline system.

System	# of output units	WER
System P	24	8.7
System G	19	17.3

Table 7.4. Performance of phoneme and grapheme baseline systems on Numbers task. The performance is measured in terms of Word Error Rate (WER) expressed in %.

Similar to isolated word recognition task, we studied grapheme-based systems where, the phonetic knowledge is used during the training, but hidden during recognition.

Grapheme-Based ASR Studies

We performed grapheme-based ASR studies by:

(a) Training a single MLP jointly modelling the phoneme and grapheme subword units ($K \times R$ output units). During recognition, the phoneme information is marginalized out according to (7.9)and decoding is performed according to (7.2).

(b) Mapping the phoneme units to values of broad-phonetic class and, training a single MLP jointly modelling grapheme units and values of broad-phonetic class. During recognition, the broad-phonetic class information is marginalized out according to (7.9)and decoding is performed according to (7.2).

Unlike the PhoneBook database, the OGI Numbers95 database has fewer phonemes ($K = 24$)and graphemes ($R = 19$). So, we trained an MLP with $24 \times 19 = 456$ output units (*System PG*). During training, at each iteration we marginalized out the phoneme information as per (7.9) and perform Viterbi decoding according to (7.2) to get the segmentation in terms of graphemes. We performed recognition experiments by marginalizing the grapheme subword units according to (7.8) and decoding according to (7.1), and similarly we performed recognition experiments by marginalizing out the phoneme subword units according to (7.9) and decoding according to (7.2). As reported in Table 7.5, there is no improvement in the performance of the phoneme system, but there is a significant improvement in the performance of the grapheme system.

Subword Unit	Subword Unit Hidden	WER
Phoneme	Grapheme	8.9
Grapheme	Phoneme	14.5

Table 7.5. Performance of phoneme-only and grapheme-only system by marginalizing (hide) grapheme and phoneme, respectively, at the output of phoneme-grapheme MLP (*System PG*) on Numbers task. The performance is measured in terms of word error rate (WER) expressed in %.

We next studied systems, where the phoneme units are mapped to the values of broad-phonetic-class representation as done earlier in the isolated word recognition studies (Section 7.4.1). The different broad-phonetic-classes along with their values are given in Table 7.2. The mapping between the phonemes and the values of the broad-phonetic-class were obtained from a *International Phonetic Alphabet (IPA) chart*. Table 7.6 shows the mapping between phonemes and the values of the broad-phonetic-classes.

Like the isolated word recognition task on PhoneBook, we studied three different grapheme-broad-phonetic-class systems corresponding to the different broad-phonetic-classes:

1. *System GBM*: The broad-phonetic class is manner.

2. *System GBP*: The broad-phonetic class is place.

3. *System GBH*: The broad-phonetic class is height.

We trained a single MLP jointly modelling the graphemes and the values of broad-phonetic class for each of the systems described above via embedded Viterbi training. We performed recognition studies by marginalizing the broad-phonetic-class according to (7.9) and, performing decoding just using grapheme transcription.

Table 7.7 presents the experimental results of this study. The grapheme systems using broad-phonetic class information as auxiliary source of knowledge perform significantly better than the grapheme baseline system. This performance is still poorer compared to phoneme-based ASR system. Similar trend was observed in case of isolated word recognition studies on PhoneBook database (see Table 7.3).

Phoneme	Word	Manner	Place	Height
h#	sil	sil	sil	sil
k	siX	stop	dorsal	max
s	Six	fricative	alveolar	max
f	Five	fricative	labial	max
th	THirty	fricative	dental	max
hh	Hundred	aspirant	unknown	max
d	hundreD	voiced stop	alveolar	max
z	Zero	voiced fricative	alveolar	max
v	fiVe	voiced fricative	labial	max
n	NiNe	nasal	alveolar	max
l	eLeven	approximant	lateral	h4
r	fouR	approximant	retroflex	h2
w	tWelve	approximant	mid	h4
iy	eightY	vowel	front	h4
ih	fIfty	vowel	front	h3
eh	sEven	vowel	front	h2
ey	Eight	vowel	front	h3
ay	fIve	vowel	mid	h3
ah	One	vowel	back	h1
ao	fOrty	vowel	back	h2
ow	zerO	vowel	back	h3
uw	twO	vowel	back	h4

Table 7.6. Mapping between the phonemes (OGIbet representation) and the values of the broad-phonetic-classes for Numbers task. sil - silence, h1 - very low height, h2 - low height, h3 - high height and h4 - very high height. The value of broad-phonetic-classes are similar to the one used in (Hosom, 2000, Chapter 7).

Phoneme-Grapheme Based ASR Studies

Starting from the improved grapheme-based ASR systems, we then studied the performance of the phoneme-grapheme system. We studied two different phoneme-based ASR systems (based on how the scaled-likelihood is estimated):

(a) Modelling the phoneme and grapheme subword units through a single MLP. For such a system the scaled-likelihood is estimated as per (7.7) from the posterior output of the MLP and the decoding is performed according to (7.5) (*System PG*).

(b) Modelling phoneme units and grapheme units through different MLPs. The scaled-likelihood $p_{sl}(x_n|q_n, l_n)$ is then obtained from the scaled-likelihood estimate of phoneme units and grapheme units according to (7.10) and, the decoding is performed according to (7.5). In our previous study on isolated word recognition task on PhoneBook (Section 7.4.1), the best results were obtained by weighting the $\log$ probability streams of phoneme and grapheme differently. However, in this study we estimated $p_{sl}(x_n|q_n, l_n)$ exactly according to (7.10).

The results of the phoneme-grapheme ASR studies are given in Table 7.8. The first row contains the performance of the phoneme baseline system. The second row contains the performance of *System PG*. This system performs poorer compared to the baseline system. The remaining rows are the results obtained for the phoneme-grapheme system where, the phoneme units and grapheme units are modelled by different MLPs. In all these systems, the grapheme scaled-likelihood is estimated by marginalizing the phonetic information according to (7.9). These systems perform better than the baseline system, especially the systems where the grapheme scale-likelihood is obtained by marginalizing out the broad-phonetic class information.

System	Broad-phonetic class	WER
System G	-	17.3
System GBM	Manner	15.4
System GBP	Place	**13.8**
System GBH	Height	13.9

Table 7.7. Performance of grapheme-based ASR system where the broad-phonetic class information is treated as auxiliary source of knowledge on Numbers task. The performance is measured in terms of Word Error Rate (WER) expressed in %. The best systems are marked in bold face.

Phoneme	Grapheme	WER
System P	-	8.7
System PG	*System PG*	9.4
System P	*System PG*	8.3
System P	*System GBM*	7.9†
System P	*System GBP*	7.9†
System P	*System GBH*	**7.8**†

Table 7.8. Performance of phoneme-grapheme system on Numbers task. Columns 1 and 2 indicate from which of the MLPs the phoneme and grapheme scaled-likelihood estimates were estimated, respectively for the system where the independence between phoneme units and grapheme units is assumed (Eq. 7.10). The performance is measured in terms of word error rate (WER) expressed in %. The best system is indicated in bold face. † indicates that the improvement in the performance of the system over the baseline is statistically significant (above 95% confidence).

Until now, we have examined the grapheme-based ASR systems with context-independent graphemes. State-of-the-art systems model context-dependent phoneme units. The task of recognition of natural numbers on OGI Numbers 95 database has a small vocabulary and, so has only 85 context-dependent graphemes. Hence, we were interested in studying the performance of a grapheme-based ASR system using context-dependent graphemes[4]. In this section, we have seen that the grapheme-based ASR performance is better when they are modelled along with phonetic information. However, training a single MLP with 24×85 output units is a difficult task. So, we took an alternate approach to estimate $P(q_n = k, l_n = r|x_n)$ as following:

$$P(q_n = k, l_n = r|x_n) = P(l_n = r|x_n, q_n = k)P(q_n = k|x_n) \tag{7.12}$$

In this system, two ANNs are trained, one with $R = 85$ output units to estimate $P(l_n = r|x_n, q_n = k)$ and one with $K = 24$ units (i.e., *System P*) to estimate $P(q_n = k|x_n)$. This approach is similar to the approach earlier proposed in the literature to model context-dependent units in the framework of hybrid HMM/ANN system (Bourlard *et al.*, 1992). Furthermore, it is important to note that this approach relaxes the conditional independence assumption made earlier in (7.3). The phoneme information in (7.12) can be marginalized out to estimate $P(l_n = r|x_n)$ in the following way:

$$P(l_n = r|x_n) = \sum_{k=1}^{k=K} P(q_n = k, l_n = r|x_n) \tag{7.13}$$

$P(l_n = r|x_n)$ can be then scaled by the prior $P(l_n = r)$ to obtain the scaled-likelihood, and used as the emission likelihood to decode in the grapheme space according to (7.2).

The ANN estimating $P(l_n = r|x_n, q_n = k)$ has phoneme information as inputs in addition to PLP feature vectors ($351+9*24 = 567$ input units). We provided posteriors obtained from *System P* as phoneme information for the contextual frames, except for the center frame. While training the ANN,

[4]This study and the context-dependent grapheme-based ASR studies reported later were jointly done with John Dines at IDIAP.

the center frame information is defined based on the knowledge of phoneme segmentation. During recognition, we define the center frame for all possible phonemes i.e., perform K forward passes and sum all the probabilities as in (7.13) to obtain $P(l_n|x_n)$. The $P(l_n|x_n)$ is then transformed into scaled-likelihood and used as emission probability to perform decoding in the context-dependent grapheme space.

We trained an hybrid HMM/ANN system in the lines of (7.12) with 85 context-dependent grapheme units as the output of ANN. This system is denoted as *System CD-G* [5]. We performed recognition just using context-dependent grapheme subword units. The performance of this system is given in Table 7.9. This system yields performance comparative to *System P* and better than the grapheme systems in Table 7.7. This is quite interesting as this recognizer is **purely** grapheme based. This motivated us to further study context-dependent grapheme-based ASR systems. We present these studies in the Section 7.5.

System	WER
System P	8.7
System G	17.3
System CD-G	8.9

Table 7.9. Performance of grapheme-based ASR system with context-dependent graphemes as subword units on Numbers task. The performance is measured in terms of word error rate (WER) expressed in %. The performances of *System P* and *System G* are repeated again for comparison.

7.4.3 Short Summary

In this section, we studied phoneme-grapheme based ASR system on two different ASR tasks. The studies show that:

- The performance of context-independent grapheme-based ASR system performance can be improved by integrating phonetic knowledge. However, even after integrating phonetic information the performance of context-independent grapheme based ASR system is poorer compared to context-independent phoneme based ASR system.
- Phoneme-grapheme based ASR system performs better than the phoneme-based ASR system.
- In phoneme-grapheme based ASR system, the approach to model the phoneme units and grapheme units through independent ANNs is better than jointly modelling them through a single ANN.

7.5 Context-Dependent Graphemes

In the previous section, we studied how to jointly model the phoneme and grapheme subword units to improve ASR performance. When studying grapheme-based ASR systems on Numbers task, we found that by modelling grapheme context information performance similar to phoneme-based ASR system can be achieved. This motivated us to further investigate the use of context-dependent graphemes for ASR, as such a system has different advantages such as, easy lexicon definition,

[5] In addition to studying this alternate approach for context-dependent graphemes, we studied it for context-independent graphemes too. This approach yielded significant improvement in the performance of the context-independent grapheme based ASR system (12.5% WER) when compared to the other context-independent grapheme based ASR systems presented earlier (see Table 7.7). However, when we performed phoneme-grapheme based ASR studies (by jointly decoding in the phoneme space and grapheme space) the improvement in the performance (8.3% WER) of the ASR system was not as significant as compared to the phoneme-grapheme based ASR systems presented earlier (see Table 7.8). The main reason behind this is that the grapheme-based ASR systems which were trained with broad-phonetic-class information provide more complimentary information compared to the grapheme-based ASR system which was trained with phoneme information (posterior probabilities obtained from the baseline ANN).

unique word representation (discussed earlier in this chapter). In this section, we present the ASR experiments just using context-dependent graphemes as subword units on:

- Numbers task: The vocabulary size is small (30 words) and so, has only 85 context-dependent graphemes.
- DARPA resourse management (RM) task: Continuous speech recognition task with a vocabulary size of 992 words.

7.5.1 Numbers Task

We studied HMM/GMM systems and hybrid HMM/ANN systems using different context-dependent subword units,

- *GMM-CD-P*: HMM/GMM system using context-dependent phonemes as subword units.
- *ANN-CD-P*: Hybrid HMM/ANN system using context-dependent phonemes as subword units.
- *GMM-CD-G*: HMM/GMM system using context-dependent graphemes as subword units.
- *ANN-CD-G*: Hybrid HMM/ANN system using context-dependent graphemes as subword units.

The HMM/GMM systems were trained with 3 emitting states per subword unit and 12 mixtures per state with 39 dimensional PLP feature vector using HTK toolkit (Young *et al.*, 1997). There were 80 context-dependent phonemes and 85 context-dependent graphemes. The hybrid HMM/ANN systems were trained via embedded Viterbi training. The parameters of the MLPs of hybrid HMM/ANN systems were same. As it can be seen from the Table 7.10 presenting the performance of these systems, the HMM/GMM system and hybrid HMM/ANN system both using only context-dependent grapheme subword units perform significantly better than their context-dependent phoneme counterparts. In the context of HMM/GMM system, it may be argued that the system *GMM-CD-G* has more parameters compared to the system *GMM-CD-P*. However, we have observed that increasing the number of parameters of the system *GMM-CD-P* does not leads to improvement over *GMM-CD-G*.

System	WER
GMM-CD-P	6.9
ANN-CD-P	6.8
GMM CD-G	6.0
ANN-CD-G	6.3
Tandem-CD-G	5.1
Tandem-CD-P	**4.9**

Table 7.10. Performance of different context-dependent subword units systems on Numbers task. The performance is measured in terms of Word Error Rate (WER) expressed in %. The best system is indicated in bold face.

In Chapter 5, we have seen that the TANDEM system yields the best performance on the same Numbers task. So, to further validate our results, we studied the context-dependent grapheme-based ASR system with in the framework of TANDEM systems. The tandem features that were used for our earlier studies presented in Chapter 5 were used to train two TANDEM systems. One with context-dependent grapheme units (*Tandem-CD-G*) and the second with context-dependent phoneme units (*Tandem-CD-P*) with the same configurations of *GMM-CD-G* and *GMM-CD-P*, respectively. The results are given in Table 7.10. It is quite interesting to note that the context-dependent grapheme-based ASR system using tandem features yields close to the state-of-the-art performance. However, the same is not the case with context-independent grapheme system. The main difference between these two systems based on graphemes is that one models the context and the other does not.

In order to understand the effect of contextual modelling in context-dependent grapheme-based ASR, we performed contextual modelling studies, where we trained systems with only preceding context or only following context. The number of preceding-context-dependent and following-context-dependent phonemes were 81 and 71 (including short pause model in HTK), respectively. The number of preceding-context-dependent and following-context-dependent graphemes were 75 and 68, respectively. All the systems were trained using HTK toolkit with 3 emitting states per subword unit and 12 mixtures per state. The results of this study are given in Table 7.11. The results indicate that the effect of modelling context in grapheme-based system is similar to that of modelling context in phoneme-based system. In other words, the context-dependent grapheme units behave like phoneme units. This could be possibly the reason why *System CD-G* reported in the previous section yields relatively lower performance compared to *ANN-CD-G*; as we were feeding the phoneme information from *System P* as additional input to *System CD-G*, which can be noisy.

Subword unit	Context	Feature	WER
Phoneme	Following	PLP	9.1
Phoneme	Preceding	PLP	13.5
Grapheme	Following	PLP	9.6
Grapheme	Preceding	PLP	14.1
Phoneme	Following	TANDEM	5.2
Phoneme	Preceding	TANDEM	6.8
Grapheme	Following	TANDEM	6.6
Grapheme	Preceding	TANDEM	9.5

Table 7.11. Results of phoneme and grapheme contextual modelling studies on Numbers task. The performance is measured in terms of Word Error Rate (WER) expressed in %.

One of the key difference between context-dependent grapheme and context-dependent phoneme is that noisy phoneme transcription is relied upon for the phoneme-based system. Also, the main idea behind modelling context in phoneme-based ASR is to capture the influence of phonemes on each other; where as in grapheme-based system, our studies suggest that by modelling context we may be able to jointly model co-articulatory effects and pronunciation variation. This could be the possible reasons why there is a significant difference between the performance of systems *GMM-CD-P* and *GMM-CD-G*, and systems *ANN-CD-P* and *ANN-CD-G*. The TANDEM system is able to handle the noise in the phoneme transcriptions (possibly due to projection of the acoustic features on speech class discriminatory dimensions) and yields state-of-the-art performances for both type of context-dependent subword units. The OGI Numbers95 task contains only 30 words and so, only a few context-dependent subword units. It can be expected that in this task there is a one-to-one correspondence between context-dependent grapheme targets to phoneme-targets, which may not be true with increasing vocabulary size. Hence, we studied context-dependent grapheme-based ASR for large vocabulary continuous speech recognition task.

7.5.2 Continuous Speech Recognition (DARPA RM) Task

We used DARPA RM corpus for large vocabulary continuous speech recognition task with context-dependent graphemes as subword unit. The acoustic feature x_n is 39 dimensional PLP cepstral features estimated every 10 ms with a frame size of 30 ms. The vocabulary contains 992 words. We defined two lexicons, namely, phoneme dictionary where the pronunciation is defined in terms of phonemes and, grapheme dictionary where the pronunciation is defined in terms of graphemes. Phoneme dictionary contains multiple pronunciation for a few words making the number of lexical entries 1032. In the grapheme dictionary, the pronunciation of digits and abbreviated words were expanded in terms of grapheme subword units, such as 1 is replaced by $/O//N//E/$ or $A-A-W$

as $/A//\_//A//\_//D//O//U//B//L//E//U/$[6]. The language model is a word pair grammar. There are 44 phonemes associated to $\mathcal{Q}$ and 29 graphemes associated to $\mathcal{L}$. The definition of the training, validation and test set can be found in Section 3.8.3.

Similar to our previous studies on Numbers task, we trained:

1. A HMM/GMM system with context-dependent phoneme acoustic models.
2. A HMM/GMM system with context-dependent graphemes acoustic models.
3. A TANDEM system with context-dependent phoneme acoustic models
4. A TANDEM system with context-dependent graphemes acoustic models.

The HMM/GMM systems used 39 dimensional PLP cepstral features as acoustic observation. We trained an MLP with 44 output units corresponding to the context-independent phonemes and. extracted the 44 dimensional tandem-features using this MLP as described in Section 3.6 of Chapter 3.

The systems were trained using HTK toolkit (Young *et al.*, 1997). The acoustic models were trained through: 8 iterations of reestimation on context-independent models, 2 iterations of reestimation on context-dependent models followed by model tying[7], 7 iterations of reestimation on tied context-dependent models and finally increment of mixtures from 1 to 8 in multiples of two with 3 iterations of reestimation at each increment step.

The recognition results of the HMM/GMM systems trained with PLP features are give in Table 7.12. The system using phoneme as subword units performs better than the system using grapheme as subword units.

Subword Unit	WER
Phoneme	**7.6**
Grapheme	10.2

Table 7.12. Recognition performance of HMM/GMM system trained on DARPA resource management corpus with context-dependent phoneme acoustic models and context-dependent grapheme acoustic models. The acoustic feature vector was 39 dimensional PLP cepstral features. The performance is measured in terms of word error rate (WER) expressed in %. The best system indicated in bold face.

The recognition results of the TANDEM systems using different subword units are give in Table 7.13. The TANDEM system performs better than the HMM/GMM system (using PLP cepstral features) for both type of subword units. Also, the amount of gain for grapheme-based system is more than the phoneme-based system making the two systems more comparable. An explanation for this can be that the TANDEM system can integrate phonetic knowledge through discriminative tandem features.

Subword Unit	WER
Phoneme	**6.8**
Grapheme	7.4

Table 7.13. Recognition performance of TANDEM system trained on DARPA resource management corpus with context-dependent phoneme acoustic models and context-dependent grapheme acoustic models. The performance is measured in terms of word error rate (WER) expressed in %. The best system is indicated in bold face.

In our earlier phoneme-grapheme studies, we have seen that by jointly decoding in phoneme state space and grapheme state space, the performance of ASR can be improved. However, in

[6] It can be noted that $/U/$ can be written as $/Y//O//U/$. However, this can lead to a Markov model of longer length which may not match well.

[7] The question set for tying consisted of singleton questions about left and right context.

large vocabulary systems with context-dependent acoustic models this is an expansive computation. One way to combine the information from these two different subword units would be to decode in each individual space and then combine the recognized word sequences by technique such as ROVER (Fiscus, 1997). Another way would be to merge the two acoustic models and dictionaries, and perform decoding in a standard way. This way the best acoustic model representation of the word is chosen at the decoding time. We chose the later approach because of its simplicity. Also, we wanted to perform analysis, such as, for what kind of words the standard ASR system would prefer a pronunciation model based on grapheme in the presence of pronunciation model based on phoneme. We performed recognition studies by merging the phoneme and grapheme acoustic models and their dictionaries. The results of this study are given in Table 7.14.

System	WER
HMM/GMM	7.4
TANDEM	**6.4**

Table 7.14. Recognition performance of HMM/GMM system using PLP features and TANDEM system trained on DARPA resource management corpus with merged acoustic models and dictionaries. The performance is measured in terms of word error rate (WER) expressed in %.

Merging of the acoustic models and dictionaries improves the ASR performance overall. We performed an analysis by counting the number of function words (e.g., to) and content words (e.g., navy) modelled during recognition by the phoneme acoustic models and grapheme acoustic models. The acoustic models trained with PLP features were used for this analysis. The result of this analysis is given in 7.15. The analysis shows that grapheme representation is more preferred when the word is a function word. Similarly, we also performed analysis in terms of length of word (number of graphemes). Table 7.16 presents the result of this analysis. The analysis shows that words short in terms of length (number of grapheme) prefer graphemes.

Type of Word	Grapheme	Phoneme
Function	1021	1520
Content	2718	5026

Table 7.15. Analysis in terms of number of function words and content words modelled during recognition by different acoustic models.

ASR studies on both Numbers task and DARPA RM task has shown that by using context-dependent grapheme as subword units in standard HMM-based ASR system performance competitive to the HMM-based ASR system using context-dependent phoneme as subword units. In both of these tasks, the words that are present in the lexicon are present in both training set and test set. In other words, during recognition there is no unseen grapheme context that the ASR system has to deal with. Further ASR studies need to be done to know how much context-dependent grapheme-based ASR systems for English language are able to generalize for unseen grapheme contexts.

7.5.3 Discussion

In this section, we studied context-dependent grapheme-based ASR for two different tasks and compared their performance with their respective standard context-dependent phoneme-based ASR system. The two different tasks that were Numbers task and DARPA RM task. We observed that for Numbers task the context-dependent grapheme-based ASR system yielded performance similar or better than the context-dependent phoneme-based ASR system. However, in case of DARPA RM task the performance of context-dependent grapheme-based ASR system was worse than the context-dependent phoneme-based ASR system. One of the main difference between the two tasks was that the Numbers task had a small vocabulary of 30 words, where as, DARPA RM task had 992 words. Since Numbers had small vocabulary, 80 context-dependent phonemes and

Length	Grapheme	Phoneme
1	15	26
2	594	906
3	817	1185
4	787	1184
5	398	737
6	363	535
7	284	647
8	246	567
9	494	1171
10	49	180
11	451	856
12	25	97
13	5	17
14	7	23
15	4	6
16	0	4
17	0	9
18	0	8
19	0	2
21	0	2
22	0	2

Table 7.16. Analysis in terms of length of word (number of graphemes) and the type of acoustic model used during recognition.

85 context-dependent graphems, there can exist to one-to-one mapping between them. However, in case of DARPA RM task there may not exist such a mapping. One way to visualize how different context-dependent phonemes were different from context-dependent graphemes is computing the mutual information between context-dependent phoneme streams and context-dependent grapheme streams. The mutual information of two random variable is a quantity that measures the statistical dependence between the two variables (Shannon and Weaver, 1963; Cover and Thomas, 1991; Papoulis, 1984).

If X and Y are two random variables then mutual information $I(X;Y)$ can be expressed as

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X) \tag{7.14}$$

where $H(X)$ is the uncertainty (entropy) about X, $H(Y)$ is the uncertainty about Y, $H(X|Y)$ is the conditional entropy (uncertainty about X given that we know Y) and $H(Y|X)$ is the conditional entropy (uncertainty about Y given that we know X). Yet another interpretation of $I(X;Y)$ is: the information that Y tells us about X is the reduction in uncertainty about X due to the knowledge of Y.

We computed mutual information between context-dependent phoneme and context-dependent grapheme streams for the two tasks in the following manner:

- Forced alignment of the training data using the trained models of context-dependent phoneme-based ASR system to obtain context-dependent phoneme stream X.
- Forced alignment of the training data using the trained models of context-dependent grapheme-based ASR system to obtain context-dependent grapheme stream Y.
- Computing the mutual information between the two streams.

$$I(X;Y) = \sum_x \sum_y p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) \tag{7.15}$$

where x and y are realizations of X and Y, respectively.

We did it for acoustic models trained with both PLP feature and tandem feature. Table 7.17 and Table 7.18 show the result of this study for Numbers task and DARPA RM task, respectively.

Feature	$I(X;Y)$	$H(X)$	$H(Y)$	$H(X\|Y)$	$H(Y\|X)$
PLP	2.868	3.578	3.579	0.710	0.711
TANDEM	2.923	3.606	3.611	0.683	0.688

Table 7.17. Mutual information between context-dependent phoneme stream and context-dependent grapheme stream for Numbers task. $I(X;Y)$ is the mutual information, $H(X)$ entropy of the phoneme stream, $H(Y)$ is the entropy of the grapheme stream, $H(X|Y)$ and $H(Y|X)$ are the conditional entropies.

Feature	$I(X;Y)$	$H(X)$	$H(Y)$	$H(X\|Y)$	$H(Y\|X)$
PLP	5.682	6.586	6.424	0.904	0.742
TANDEM	5.523	6.489	6.425	0.966	0.902

Table 7.18. Mutual information between context-dependent phoneme stream and context-dependent grapheme stream for DARPA RM task. $I(X;Y)$ is the mutual information, $H(X)$ entropy of the phoneme stream, $H(Y)$ is the entropy of the grapheme stream, $H(X|Y)$ and $H(Y|X)$ are the conditional entropies.

It can be observed from the tables that the mutual information there is certain dependency between the context-dependent phoneme stream and context-dependent grapheme stream (mutual information is above zero). This dependency is stronger in case of Numbers task then DARPA task where both streams convey information about each other almost equally well (see the conditional entropies in Table 7.17). This explains why different trends were observed in Numbers task and DARPA RM task. In case of the PLP-based system of DARPA RM task information that context-dependent phoneme conveys about the context-dependent grapheme is more than the information that context-dependent grapheme conveys about the context-dependent phoneme. In case of the TANDEM system though we see the same behaviour the difference between the conditional entropies $H(X|Y)$ and $H(Y|X)$ is low compared to PLP-based system.

7.6 Summary and Conclusion

In this chapter, we investigated the use of grapheme as subword units for ASR system in English language, where there is a weak correspondence between the written form (grapheme) and spoken form (phoneme). We proposed a phoneme-grapheme based ASR where, during the training, acoustic models for both phoneme and grapheme subword units are trained. During recognition, the decoding is done either jointly in both the subword unit spaces or in one of the subword unit space. We studied this system in the framework of hybrid HMM/ANN based ASR system. Recognition studies performed on different ASR tasks show that:

- Though the performance of context-independent grapheme-based ASR system can be improved by using phonetic knowledge, the perform of context-independent grapheme-based system is worse than the context-independent phoneme-based ASR system.
- Phoneme-grapheme based ASR system performs better than the standard phoneme-based ASR system.

While investigating phoneme-grapheme based ASR system, we observed that by modelling subword contexts the performance of grapheme-based ASR system can be improved to that of phoneme-based ASR system. This motivated us to investigate the use of context-dependent graphemes for ASR. ASR studies performed in the framework of HMM/GMM system and TANDEM system using context-dependent subword units show that:

- Context-dependent graphemes behave like phoneme units.
- In standard ASR, by using context-dependent graphemes as subword units performance competitive to the state-of-the-art context-dependent phoneme-based ASR system can be achieved.
- TANDEM system performs better than the standard cepstral feature based HMM/GMM system, even for context-dependent grapheme-based ASR system.
- An ASR system using both phoneme and grapheme subword units can perform better than the state-of-the-art ASR context-dependent phoneme-based ASR system.

Chapter 8

Summary and Conclusion

The central theme of the present thesis was the investigation of the value and usage of auxiliary knowledge sources to improve state-of-the-art ASR systems. More specifically, we investigated different knowledge sources related to:

1. Acoustic features (auxiliary features).
2. Pronunciation models (auxiliary subword units).

As discussed in the thesis, integrating auxiliary knowledge sources in standard HMM-based ASR systems is not an obvious problem, for instance:

- In order to integrate auxiliary features in standard ASR system, the optimal solution does not necessarily consist in simply appending them as additional features to the standard acoustic features.
- The auxiliary feature may not be always available or a reliable estimate may not be available. In such a case, we may want to hide the auxiliary features..
- In order to integrate auxiliary subword units, we need to jointly model two different subword word units (instead of only one) along with acoustic feature sequence and train models for both the subwords. During recognition, we may want use either both or one.

8.1 Auxiliary Features

The auxiliary features that were examined in this thesis were (1) pitch frequency, (2) short-term energy, and (3) rate-of-speech. The auxiliary features were directly extracted from the speech signal. We studied different ways to integrate auxiliary features in standard HMM-based ASR system:

1. Appending the auxiliary feature to the standard acoustic features.
2. Conditioning the emission distribution upon the auxiliary features.

Based upon the experiments conducted on different ASR tasks in the framework of hybrid HMM/ANN system, HMM/DBN-GMM system and TANDEM systems, we draw the following conclusions:

- The performance of state-of-the-art ASR system can be improved by integrating auxiliary features pitch frequency, short-term energy and rate-of-speech in standard HMM-based ASR system. In CTS task, though there was no improvement, there was not a degradation in the performance either.

- In ASR systems with context-independent subword units, it is better to condition the emission distribution upon the auxiliary feature when using standard cepstral features.
- In ASR systems with context-dependent subword units, concatenation of the auxiliary feature pitch frequency, short-term energy and rate-of-speech to the standard cepstral feature or tandem features helps in improving the performance.
- Integrating auxiliary feature in TANDEM system improves the performance of ASR in both clean and noisy conditions.

8.2 Auxiliary Subword Units

We extended the approach to integrate auxiliary sources of knowledge to jointly model more than one subword units. We proposed a phoneme-grapheme based ASR system that during training jointly models the phoneme subword units and the grapheme subword units. During recognition, the decoding is done either using one or both the subword units. In doing so, we used grapheme as auxiliary subword units. We studied the phoneme-grapheme based ASR system in the framework of hybrid HMM/ANN system for different ASR tasks on English language. In addition to investigating phoneme-grapheme based ASR system, we also investigated a grapheme-based ASR system where, the phonetic information is only used during training and, during recognition the phonetic information is marginalized. The main conclusions from this work are:

- Context-independent grapheme-based ASR system performs worse than the context-independent phoneme-based ASR system.
- The phoneme-grapheme based ASR system performs better than the standard phoneme-based ASR system.

In phoneme-grapheme based ASR system studies, we observed that it is beneficial to model the grapheme contextual information similar to phonemes. We investigated the context-dependent grapheme-based ASR system for two different ASR tasks. The main conclusions drawn from this study are the following:

- Context-dependent grapheme units behave like phoneme units.
- ASR systems using context-dependent grapheme as subword units can yield performance competitive to the ASR systems using context-dependent phoneme as subword units.
- Similar to phoneme-grapheme based ASR system, an ASR system using both phoneme and grapheme subword units can perform better than context-dependent grapheme-based ASR system and standard context-dependent phoneme-based ASR system.

8.3 Model Evaluation

We proposed an approach to evaluate the adequacy of the baseform pronunciation of words. This approach is based on relaxing lexical constraints in the baseform pronunciation pronunciation, inferring new pronunciation variants and, measuring the stability of the baseform pronunciation by evaluating the inferred pronunciation variants by confidence measure and Levenshtein distance. The proposed approach was used to compare the quality of different acoustic models and, select new pronunciation variants for the lexicon. The main conclusions of this work are:

- Integrating auxiliary features improves the stability of the baseform pronunciation, i.e., it improves the matching and discriminating properties of baseform pronunciation.
- The ASR system can be improved by the selecting pronunciation variants using the proposed approach.

8.4 Future Directions

There are different research directions that could be followed in the future, including:

- The auxiliary features modelled in HMM/DBN-GMM framework the mixture variable has been shared between the standard feature and auxiliary feature (see Figure 4.2). It would be interesting to have a separate mixture variable for auxiliary feature such as two mixture components for pitch frequency (voiced and unvoiced).

- In this thesis, we have studied integration of a single auxiliary feature at a time. It would be interesting to model multiple auxiliary features and their relation with each other. Apart from the concatenation and conditioning approach, this can be investigated in the multi-stream framework.

- The auxiliary features studied in this thesis were extracted at segmental level (i.e. 30 ms). The auxiliary features can convey different information at different levels. For instance, the variation of pitch frequency or energy over time (suprasegmental level) carry prosody information. These information can be modelled in a hierarchical or layered fashion (Bourlard *et al.*, 2004). TANDEM approach is one such layered approach. The information carried by auxiliary feature at segmental level is integrated in the lower layer (e.g., at the level of subword units modelling). The suprasegmental information carried by the auxiliary features can be further modelled at the next higher level (e.g. word level, sentence level).

- In this thesis, we observed that by modelling auxiliary features we can improve the matching between the acoustic observation and the pronunciation model. An alternate way to pronunciation modelling would be to fix the pronunciation of the words and, then identifiying and integrating auxiliary features that can improve the matching and discriminating properties of single pronunciation.

Bibliography

Andreou, A., Kamm, T., and Cohen, J. (1994). Experiments in vocal tract normalization. In *Proceedings of the CAIP Workshop: Frontiers in Speech Recognition*.

Atal, B. S. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *J. Acoust. Soc. Amer.*, **55**(6), 1304–1312.

Athineos, M., Hermansky, H., and Ellis, D. P. (2004). Plp^2: Autoregressive modeling of auditory-like 2-D spectro-temporal patterns. In *Workshop on Statistical and Perceptual Audio Processing (SAPA)*.

Bacchiani, M. and Ostendorf, M. (1999). Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, **29**(2–4), 99–114.

Bagshaw, P. C., Hiller, S. M., and Jack, M. A. (1993). Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. In *Proceedings of the 3rd European Conference on Speech Communication and Technology (Eurospeech '93)*, pages 1003–1006, Berlin.

Bahl, L. R., Bakis, R., Jelinek, F., and Mercer, R. L. (1980). Language-model/acoustic-channel-model balance mechanism. *IBM Technical Disclosure Bulletin*, **23**(7B), 3464–3465.

Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **PAMI-5**(2), 179–190.

Baker, J. K. (1975). The DRAGON system-an overview. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-23**(1), 24–29.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximisation technique in the statistical analysis of probabilistic functions of Markov chains. *Ann. of Math. Stat.*, **41**(1), 164–171.

Bengio, Y., de Mori, R., Flammia, G., and Kompe, R. (1992). Neural network - Gaussian mixture hybrid for speech recognition or density estimation. In J. Moody, S. Hanson, and R. Lipmann, editors, *Advances in Neural Information Processing Systems, NIPS 4*, pages 175–182. Morgan Kauffman.

Beyerlein, P. (1998). Discriminative model combination. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 481–484, Seattle, USA.

Bilmes, J. and Zweig, G. (2002). The graphical models toolkit: An open source software system for speech and time-series processing. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages IV:3916–3919, Orlando Florida.

Bilmes, J. A. (1997). A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, ICSI, Berkeley.

Bilmes, J. A. (1999). *Natural Statistical Models for Automatic Speech Recognition*. PhD dissertation, Dept. of EECS, CS Division, U.C. Berkeley, California.

Bilmes, J. A. (2004). Graphical models and automatic speech recognition. In M. Johnson, S. P. Khudanpur, M. Ostendorf, and R. Rosenfeld, editors, *Mathematical foundations of speech and language processing*, volume 138 of *The IMA volumes in mathematics and application*, pages 191–245. Springer-Verlag, NewYork.

Bisani, M. and Ney, H. (2004). Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 409–412, Montreal Canada.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford university press, Walten street, Oxford.

Bourlard, H. and Morgan, N. (1994). *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers.

Bourlard, H., Morgan, N., Wooters, C., and Renals, S. (1992). CDNN: A context dependent neural network for continuous speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages II–349–II–352, San Francisco, USA.

Bourlard, H., Bengio, S., Magimai.-Doss, M., Zhu, Q., Mesot, B., and Morgan, N. (2004). Towards using hierarchical posteriors for flexible automatic speech recognition systems. In *Proceedings of DARPA RT-04 Workshop*.

Bridle, J. S. and Cox, S. J. (1991). RecNorm: Simultaneous normalization and classification applied to speech recognition. In *Advances in Neural Information Processing Systems*, pages 234–240.

Bridle, J. S., Brown, M. D., and Chamberlain, R. M. (1983). Continuous connected word recognition using whole word templates. *The Radio and Electronic Engineer*, **53**(4), 167–175.

Cetin, O. and Ostendorf, M. (2005). Multi-rate and variable-rate modeling of speech at phone and syllable time scales. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:665–668, Philadelphia, USA.

Cole, R. A., Fanty, M., Noel, M., and Lander, T. (1994). Telephone speech corpus development at CSLU. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 1815–1818, Yokohama, Japan.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley, New York.

Dalsgaard, P., Andersen, O., and Barry, W. (1991). Multi-lingual label alignment using acoustic-phonetic features derived by neural network technique. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 197–200, Toronto, Canada.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in countinuously spoken sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-28**(4), 357–366.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, **39**, 1–38.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Recognition*. John Wiley & Sons Inc.

Dupont, S. (2000). *Etude et développement d'architectures multi-bandes et multi-modales pour la reconnaissance robuste de la parole.* Ph.D. thesis, Faculté Polytechnique de Mons, Mons, Belgium.

Dupont, S., Bourlard, H., Deroo, O., Fontaine, V., and Boite, J.-M. (1997). Hybrid HMM/ANN systems for training independent tasks: Experiments on 'PhoneBook' and related improvements. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1767–1770, Munich, Germany.

Efron, B. and Tibshirani, M. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, NewYork.

Ellis, D. P. W., Singh, R., and Sivadas, S. (2001). Tandem acoustic modeling in large-vocabulary recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:517–520, Salt Lake City, USA.

Faria, A. (2003). Pitch-based vocal tract length normalization. Technical Report TR-03-01, ICSI, Berkeley, USA.

Fiscus, J. (1997). A post-processing system to yield to yield reduced word error rates: recognized output voting error reduction (ROVER). In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pages 347–352, Santa Barbara, USA.

Forney, G. D. (1973). The Viterbi algorithm. *Proc. IEEE*, **61**(3), 263–278.

Fosler-Lussier, J. E. (1999). *Dynamic Pronunciation Models for Automatic Speech Recognition*. PhD dissertation, ICSI, University of California, Berkeley.

Fujinaga, K., Nakai, M., Shimodaira, H., and Sagayama, S. (2001). Multiple-regression hidden Markov model. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 513–516, Salt Lake City, USA.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-29**(2), 254–272.

Furui, S. (1986). Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-34**(1), 254–272.

Gales, M. J. F. (1999). Semi-tied covariance matrices for hidden Markov models. *IEEE Trans. Speech, Audio Processing*, **7**(3), 272–281.

Ganapathiraju, J., Hamekar, M., Ordowski, G., Doddington, G., and Picone, J. (2001). Syllable-based large vocabulary continuous speech recognition. *IEEE Trans. Speech, Audio Processing*, **9**(4), 358–366.

Gauvain, J. L. and Lee, C. H. (1994). Maximum a-posteriori estimation of multivariate Gaussian mixture observations. *IEEE Trans. Speech, Audio Processing*, **2**, 291–298.

Gevins, A. S. and Morgan, N. H. (1984). Ignorance-based systems. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 166–169, San Diego, USA.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial hidden Markov models. *Machine Learning*, **29**, 245–273.

Gillick, L. and Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 532–535, Glasgow, Scotland.

Godin, C. and Lockwood, P. (1989). DTW techniques for continuous speech recognition: a unified overview. *Computer Speech and Language*, **3**(2), 169–198.

Gold, B. and Morgan, N. (2000). *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley and Sons.

Haeb-Umbach, R. and Ney, H. (1992). Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1:13–16, San Francisco, USA.

Hagen, A. (2001). *Robust speech recognition based on multi-stream processing*. PhD dissertation, EPFL, Lausanne, Switzerland.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons Inc., New York.

Hazen, T. J. (1998). *The Use of Speaker Correlation Information for Automatic Speech Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Hennebert, J., Ris, C., Bourlard, H., Renals, S., and Morgan, N. (1997). Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech '97)*, pages 1951–1954, Rhodes, Greece.

Hermansky, H. (1990). Perceptual linear predictive(PLP) analysis of speech. *J. Acoust. Soc. Amer.*, **87**(4), 1738–1752.

Hermansky, H. (1999). Mel cepstrum, deltas, double-deltas,... -What else is new. In *Robust methods for speech recognition in adverse conditions*, Tampere, Finland.

Hermansky, H. (2003). TRAP-TANDEM: Data-driven extraction of temporal features from speech. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pages 255–260, U.S. Virgin Island.

Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Trans. Speech, Audio Processing*, **2**(4), 578–589.

Hermansky, H. and Sharma, S. (1998). TRAPS classifiers of temporal patterns. In *Proceedings of Int. Conf. Spoken Language Processing*, Sydney, Australia.

Hermansky, H., Fujisaki, H., and Sato, Y. (1983). Analysis and synthesis of speech based on spectral transform linear predictive method. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 777–780, Boston.

Hermansky, H., Ellis, D., and Sharma, S. (2000). Tandem connectionist feature stream extraction for conventional HMM systems. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages III–1635–1638, Istanbul, Turkey.

Hess, W. (1983). *Pitch Determination of Speech Signals: Algorithms and Devices*. Springer-Verlag, Berlin.

Holmes, W. J. and Huckvale, M. (1994). Why have HMMs been so successful for automatic speech recognition and how might they be improved? *Speech, Hearing and Language, UCL Work in Progress*, **8**, 207–219.

Hosom, J.-P. (2000). *Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information*. PhD dissertation, CSLU, OGI, USA.

Ikbal, S., Hermansky, H., and Bourlard, H. (2003a). Nonlinear Spectral Transformations for Robust Speech Recognition. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop 2003*, pages 393–398, U.S. Virgin Islands.

Ikbal, S., Misra, H., and Bourlard, H. (2003b). Phase autocorrelation (PAC) derived robust speech features. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 2, pages 133–136, Hong Kong.

Ikbal, S., Magimai.-Doss, M., Misra, H., and Bourlard, H. (2004a). Spectro-Temporal Activity Pattern (STAP) Features for Noise Robust ASR. In *Proceedings of Int. Conf. Spoken Language Processing (INTERSPEECH-ICSLP-04)*, Jeju Island, Korea.

Ikbal, S., Misra, H., Sivadas, S., Hermansky, H., and Bourlard, H. (2004b). Entropy based combination of tandem representations for robust speech recognition. In *Proceedings of Int. Conf. Spoken Language Processing (INTERSPEECH-ICSLP-04)*, Korea.

IPA-Chart (1996). http://www2.arts.gla.ac.uk/ipa/fullchart.html.

Jelinek, F. (1969). Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, **13**(6), 675–685.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proc. IEEE*, **64**, 532–556.

Jelinek, F. (1997). *Statistical Methods for Speech Processing*. Language, Speech and Communication Series MIT Press, Cambridge, MA.

Junqua, J.-C. (1993). The Lombard reflex and its role on human listeners and automatic speech recognizers. *J. Acoust. Soc. Amer.*, **93**(1), 510–524.

Kanedera, N., Hermansky, H., and Arai, T. (1998). Desired characteristics of modulation spectrum for robust automatic speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 613–616, Seattle WA, USA.

Kanthak, S. and Ney, H. (2002). Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 845–848, Orlando, USA.

Katz, S. M. (1987). Estimation of probabilities from sparse data for language model component of a speech recognizer. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-35**(3), 400–401.

Keller, M., Mariéthoz, J., and Bengio, S. (2004). Significance Tests for *bizarre* Measures in 2-Class Classification Tasks. IDIAP-RR 34, IDIAP.

Kessens, J., Wester, M., and Strik, H. (1999). Improving the performace of Dutch CSR by modelling within-word and cross-word pronunciation variation. *Speech Communication*, **29**(2–4), 193–207.

Kessens, J., Cucchiarani, C., and Strik, H. (2003). A data-driven method for modeling pronunciation variation. *Speech Communication*, **40**(4), 517–534.

Killer, M., Stüker, S., and Schultz, T. (2003). Grapheme based speech recognition. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech '03)*, pages 3141–3144, Geneva, Switzerland.

King, S. and Taylor, P. (2000). Detection of phonological features in continuous speech using neural networks. *Computer Speech and Language*, **14**(4), 333–353.

Kingsbury, B., Morgan, N., and Greenberg, S. (1998). Robust speech recognition using modulation spectrogram. *Speech Communication*, **25**, 117–132.

Klatt, D. H. (1977). Review of the ARPA speech understanding project. *J. Acoust. Soc. Amer.*, **62**(6), 1345–1366.

Konig, Y., Morgan, N., and Chandra, C. (1991). GDNN: A gender-dependent neural network for continuous speech recognition. Technical Report TR-91-071, ICSI, Berkeley, Berkeley, Califronia, USA.

Kurimo, M. (1997). *Using Self-Organizing Maps and Learning Vector Quantization for Mixture Density Hidden Markov Models*. PhD dissertation, Faculty of Information Technology in the Department of Computer Science, Helsinki University of Technology, Espoo, Finland.

Kuwabara, H. (1997). Acoustic and perceptual properties of phonemes in continuous speech as a function of speaking rate. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech '97)*, volume 2, pages 1003–1006, Rhodes, Greece.

Lauritzen, S. L. and Jensen, F. (2001). Stable local computations with conditional gaussian distributions. *Statistics and Computing*, **11**(2), 191–203.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (1998). Effiecient BackProp. In G. N. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, chapter 1, pages 9–50. Springer-Verlag.

Lee, K.-F. (1990). Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-38**(4), 599–609.

Lee, L. and Rose, R. (1996). Speaker normalization using efficient frequency warping procedures. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1:353–356, Atlanta, USA.

Leggetter, C. J. and Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, **9**, 171–185.

Lesser, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R. (1975). Organization of the Hearsay-II speech understanding system. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-23**, 11–23.

Levinson, S. E. (1985). Structural methods in automatic speech recognition. *Proc. IEEE*, **73**(11), 1625–1650.

Liporace, L. A. (1982). Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Trans. Inform. Theory*, **IT-28**(5), 729–734.

Livescu, K., Glass, J., and Bilmes, J. (2003). Hidden feature models for speech recognition using dynamic bayesian networks. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech)*, pages 2529–2532, Geneva.

Ljolje, A. (1994). High accuracy phone recognition using context-clustering and quasitriphonic models. *Computer Speech and Language*, **8**, 129–151.

Logan, B. and Moreno, P. J. (1997). Factorial hidden Markov models for speech recognition: Preliminary experiments. Technical Report Series CRL 97/7, Cambridge Research Laboratory, Massachusetts, USA.

Lowerre, B. (1976). *The Harpy Speech Recognition System*. PhD dissertation, Carnegie Melon University, Pittsburgh, USA.

Magimai-Doss, M. and Bourlard, H. (2001). Pronunciation models and their evaluation using confidence measures. Technical Report RR-01-29, IDIAP, Martigny, Switzerland.

Magimai.-Doss, M. and Bourlard, H. (2005). On the adequacy of baseform pronunciations and pronunciation variants. Lecture Notes in Computer Science No. 3361, pages 209–222. Springer-Verlag.

Magimai.-Doss, M., Stephenson, T. A., and Bourlard, H. (2003a). Using pitch frequency information in speech recognition. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech '03)*, pages 2525–2528, Geneva.

Magimai.-Doss, M., Stephenson, T. A., Bourlard, H., and Bengio, S. (2003b). Phoneme-Grapheme based automatic speech recognition system. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop 2003*, pages 94–98.

Magimai.-Doss, M., Bengio, S., and Bourlard, H. (2004a). Joint decoding for phoneme-grapheme continuous speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I–177–I–180, Montreal, Canada.

Magimai.-Doss, M., Stephenson, T. A., Ikbal, S., and Bourlard, H. (2004b). Modelling auxiliary features in tandem systems. In *Proceedings of Int. Conf. Spoken Language Processing (INTERSPEECH-ICSLP-04)*, South Korea.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proc. IEEE*, **63**, 561–580.

Makhoul, J. and Schwartz, R. (1985). Ignorance modeling. In J. S. Perkell and D. H. Klatt, editors, *Variability and Invariance in Speech Processes*. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Markel, J. D. (1972). The SIFT algorithm for fundamental frequency estimation. *IEEE Trans. Audio and Electroacoustics*, **20**, 367–377.

Markel, J. D. and Gray, A. H. (1976). *Linear prediction of speech*. Springer-Verlag, New York.

Martinez, F., Tapias, D., and Alvarez, J. (1998). Towards speech rate independence in large vocabulary continuous speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 725–728, Seattle, USA.

McCowan, I., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., and Bourlard, H. (2005). On the use of information retrieval measure for speech recognition evaluation. IDIAP-RR 73, IDIAP.

McNemar, I. (1947). Note on the sampling error of the difference between correlated proportions or percentage. *Psychometrika*, **12**, 153–157.

Merhav, N. and Ephraim, Y. (1991a). Hidden Markov modelling using the most likely state sequence. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1:469–472, Toronto, Canada.

Merhav, N. and Ephraim, Y. (1991b). Hidden Markov modeling using a dominant state sequence with application to speech recognition. *Computer Speech and Language*, **5**(4), 327–339.

Mirghafori, N. and Morgan, N. (1998). Combining connectionist multi-band and full-band probability streams for speech recognition of natural numbers. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 743–746, Sydney, Australia.

Mirghafori, N., Fosler, E., and Morgan, N. (1995). Fast speakers in large vocabulary continuous speech recognition: analysis & antidotes. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech '95)*, pages 491–494, Madrid, Spain.

Misra, H., Bourlard, H., and Tyagi, V. (2003). New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages II–741–II–744, HongKong.

Mokbel, H. and Jouvet, D. (1998). Derivation of the optimal phonetic transcription set for a word from its acoustic realisation. In *Proceedings of Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pages 73–78.

Moore, B. C. J. (1995). *Hearing.* Academic Press, San Diego.

Moore, B. C. J. (1997). *An Introduction to the Psychology of Hearing.* 4th Edition Academic Press, San Diego.

Morgan, N. and Bourlard, H. (1995). Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach. *IEEE Signal Processing Magazine*, pages 25–42.

Morgan, N. and Fosler-Luisser, E. (1998). Combining multiple esitmators of speaking rate. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 729–732, Seattle, USA.

Morgan, N., Fosler, E., and Mirghafori, N. (1997). Speech recognition using on-line estimation of speaking rate. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech '97)*, volume 4, pages 2079–2082, Rhodes, Greece.

Morris, A., Maier, V., and Green, P. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Proceedings of Int. Conf. Spoken Language Processing (INTERSPEECH-ICSLP-04)*, Jeju Island, South Korea.

Nádas, A. (1984). Estimation of the probabilities in the language model of the IBM speech recognition system. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-32**, 859–861.

Nagarajan, T., Murthy, H. A., and Hegde, R. M. (2003). Segmentation of speech into syllable-like units. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech '03)*, volume 4, pages 2893–2896, Geneva, Switzerland.

Ney, H. (1984). The use of a one–stage dynamic programming algorithm for connected word recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-32**(3), 263–271.

Ney, H. and Ortmanns, S. (2000). Progress in dynamic programming search. *IEEE Signal Processing Magazine*, **88**, 1224–1240.

O'Brien, S. M. (1993). Knowledge-based systems in speech recognition: a survey. *Int. J. Man-Machine Studies*, **38**, 71–95.

Odell, J. J. (1995). *The use of context in large vocabulary continuous speech recognition*. Ph.D. thesis, Queens College, University of Cambridge.

O'Shaughnessy, D. (1987). *Speech Communication - human and machine.* Addison Wesley.

O'Shaughnessy, D. (2003). Interacting with the computers by voice: automatic speech recognition and synthesis. *Proc. IEEE*, **91**(9), 1272–1305.

Ostendorf, M. (1999). Moving beyond the 'Beads-on-a-String' model of speech. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pages 79–83, Colorado, USA.

Ostendorf, M., Digalakis, V., and Kimball, O. (1996). From HMM's to segment models: a unified view of stochastic modelling of speech recognition. *IEEE Trans. Speech, Audio Processing*, **4**, 360–378.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes.* Second Edition McGraw-Hill, New York.

Pitrelli, J. F., Fong, C., Wong, S. H., Spitz, J. R., and Leung, H. C. (1995). PhoneBook: A phonetically-rich isolated-word telephone-speech database. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1767–1770, Detroit, USA.

Plante, F., Meyer, G. F., and Ainsworth, W. A. (1995). A pitch extraction reference database. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech '95)*, pages 837–840, Madrid, Spain.

Price, P. J., Fisher, W., and Bernstein, J. (1988). A database for continuous speech recognition in a 1000 word domain. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1:651–654, New York City, USA.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**(2), 257–286.

Rabiner, L. R. and Juang, H. W. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs New Jersey.

Reddy, D. R. (1967). Computer recognition of connected speech. *J. Acoust. Soc. Amer.*, **42**, 329–347.

Renals, S. (1988). Radial basis function network for speech pattern classification. *Electronic Letters*, **25**, 437–439.

Richard, M. D. and Lippman, R. P. (1991). Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, **3**, 461–483.

Robinson, A. J. and Fallside, F. (1991). A recurrent error propogation speech recognition system. *Computer Speech and Language*, **5**, 259–274.

Rumelhart, D. E., Hinton, G. E., and Williams, R. T. (1988). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP research group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, volume 1, pages 318–362. MA: MIT press. Reprinted in Anderson and Rosenfield, Cambridge.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-26**(1), 43–49.

Samudravijaya, K., Singh, S. K., and Rao, P. (1998). Pre-recognition measures of speaking rate. *Speech Communication*, **24**(1), 73–84.

Sankoff, D. and Kruskal, J. (1999). *Time Warps, String Edits and Macromolecules: The theory and practise of sequence comparison*. CSLI Publications, Leland Stanford Junior University.

Sarclar, M. (2000). *Pronunciation modeling for conversational speech recognition*. PhD dissertation, CSLU, Johns Hopkins University, Baltimore, USA.

Schukat-Talamazzini, E. G., Niemann, H., Eckert, W., Kuhn, T., and Rieck, S. (1993). Automatic speech recognition without phonemes. In *Proceedings of the 3rd European Conference on Speech Communication and Technology (Eurospeech '93)*, pages 129–132, Berlin.

Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J. (1985). Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1205–1208, Tampa, USA.

Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, **1**, 145–168.

Shannon, C. E. and Weaver, W. (1963). *The Mathematical Theory of Communication*. Univ of Illinois Press.

Siegler, M. A. (1995). *Measuring and Compensating for the Effects of Speech Rate in Large Vocabulary Continuous Speech Recognition*. MS dissertation, Carnegie Mellon University, Department of Electrical and Computer Engineering.

Siivola, V., Hirsimäki, T., Creutz, M., and Kurimo, M. (2003). Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of the 8th European Conference on Speech Technology and Communication (Eurospeech '03)*, pages 2293–2296, Geneva, Switzerland.

Singer, H. and Sagayama, S. (1992). Pitch dependent phone modelling for HMM based speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:273–276, San Francisco, USA.

Singh, R., Raj, B., and Stern, R. (2002). Automatic generation of sub-word units for speech recognition systems. *IEEE Trans. Speech, Audio Processing*, **10**(2), 89–99.

Sivadas, S. and Hermansky, H. (2004). On use of task independent training data in tandem feature extraction. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:541–544, Montreal, Canada.

Snedecor, G. W. and Cochran, W. G. (1989). *Statistical Methods*. Iowa State University Press, Iowa, USA.

Stephenson, T. A. (2003). *Speech recognition with auxiliary information*. PhD dissertation, Swiss Federal Institutue of Technology (EPFL), Lausanne.

Stephenson, T. A., Magimai.-Doss, M., and Bourlard, H. (2001). Modeling auxiliary information in Bayesian network based ASR. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech '01)*, pages 2765–2768, Aalborg, Denmark.

Stephenson, T. A., Magimai-Doss, M., and Bourlard, H. (2002). Mixed Bayesian networks with auxiliary variables for automatic speech recognition. In *International Conference on Pattern Recognition (ICPR 2002)*, volume 4, pages 293–296, Quebec City, PQ, Canada.

Stephenson, T. A., Magimai.-Doss, M., and Bourlard, H. (2004). Speech recognition with auxiliary information. *IEEE Trans. Speech and Audio Processing*, **12**(3), 189–203.

Stevens, S. S. (1957). On the psychophysical law. *Pyschol. Rev*, **64**, 153–181.

Stolcke, A. and Omohundro, S. M. (1994). Best-first model merging for hidden Markov models. Technical Report TR-94-003, ICSI, Berkeley, Califronia, USA.

Strik, H. and Cucchiarini, C. (1999). Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*, **29**, 225–246.

Takeda, K., Ogawa, A., and Itakura, F. (1998). Estimating entropy of language from optimal word insertion penalty. In *Proceedings of Int. Conf. Spoken Language Processing*.

Tokuda, K., Zen, H., and Kitamura, T. (2003). Trajectory modelling based on HMMs with explicit relationship between static and dynamic features. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech '03)*, Geneva, Switzerland.

Tyagi, V., McCowan, I., Bourlard, H., and Misra, H. (2003). Mel-cepstrum modulation spectrum (MCMS) features for robust ASR. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop 2003*, pages 399–404, U.S. Virgin Islands.

Varga, A., Steeneken, H., Tomlinson, M., and Jones, D. (1992). The NOISEX-92 study on the effect of additive noise on automatic speech recognition. Technical report, DRA Speech Research Unit, Malvern, England.

Velichko, Z. M. and Zagoruyko, N. G. (1970). Automatic recognition of 200 words. *Int. J. Man-Machine Studies*, **2**, 222–234.

Vergin, R., Farhat, A., and O'Shaugnessy, D. (1996). Robust gender dependent acoustic phonetic modelling in continuous speech recognition based on a new automatic male/female classification. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 1081–1084, Philadelphia, USA.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Inform. Theory*, **IT-13**, 260–269.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-37**(3), 328–339.

Wang, C. and Seneff, S. (2001). Lexical stress modeling for improved speech recognition of spontaneous telephone speech in the JUPITER domain. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech '01)*, volume 4, pages 2761–2764, Aalborg, Denmark.

Wester, M., Frankel, J., and King, S. (2004). Asynchronous articulatory feature recognition using dynamic Bayesian networks. In *Proc. IEICI Beyond HMM Workshop*, Kyoto.

Williams, G. and Renals, S. (1999). Confidence measures from local posterior probability estimates. *Computer Speech and Language*, **13**, 395–411.

Woodland, P. C. (2001). Speaker adaptation for continuous density HMMs: A Review. In *Proceedings of ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, pages 11–19, Sophia-Antipolis, France.

Young, S., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1997). Hidden Markov model toolkit V2.1 reference manual. Technical report, Speech group, Engineering Department, Cambridge University, UK.

Young, S. J. (1992). The general use of tying in phoneme-based HMM speech recognisers. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 569–572, San Francisco CA.

Zhang, Y., Diao, Q., Huang, S., Hu, W., Bartels, C., and Bilmes, J. (2003). DBN based multi-stream models for speech. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:884–887, Hong Kong.

Zhu, Q., Chen, B., Morgan, N., and Stolcke, A. (2004). On using MLP features in LVCSR. In *Proceedings of Int. Conf. Spoken Language Processing (INTERSPEECH-ICSLP-04)*, Jeju Island, South Korea.

Zue, V. W. (1985). The use of speech knowledge in automatic speech recognition. *Proc. IEEE*, **73**(11), 1602–1615.

Zweig, G., Bilmes, J., Richardson, T., Filali, K., Livescu, K., Xu, P., Jackson, K., Brandman, Y., Sandness, E., Holtz, E., Torres, J., and Byrne, B. (2002). Structurally discriminative graphical models for automatic speech recognition: Results from the 2001 Johns Hopkins summer workshop. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages I:93–96, Orlando Florida.

Zweig, G. G. (1998). *Speech Recognition with Dynamic Bayesian Networks*. PhD dissertation, University of California, Berkeley.