A GREEK BROADCAST NEWS TRANSCRIPTION SYSTEM

By
Orfeas Tsergoulas

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
TECHNICAL UNIVERSITY OF CRETE
CHANIA, GREECE
NOVEMBER 2009

TECHNICAL UNIVERSITY OF CRETE
DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**A Greek Broadcast News Transcription System**" by **Orfeas Tsergoulas** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: November 2009

Supervisor: _____
Prof. Vasilis Digalakis

Readers: _____
Assoc. Prof. Alexandros Potamianos

_____
Prof. Athanasios Liavas

TECHNICAL UNIVERSITY OF CRETE

Date: **November 2009**

Author: **Orfeas Tsergoulas**

Title: **A Greek Broadcast News Transcription System**

Department: **Electronics and Computer Engineering**

Degree: **M.Sc.** Convocation: **November** Year: **2009**

Permission is herewith granted to Technical University of Crete to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

To my parents

Table of Contents

List of Tables

List of Figures

Abstract

Recent advances in speech information processing, natural language processing, statistical learning theory and computer technology lead to major progress in speech recognition technology and put various speech recognition applications such as voice-driven car navigation systems, automated call center, and desktop speech recognition software, to practical use.

In this work we present the development and implementation of a Greek broadcast news system. This system is based on a continuous mixture density, tied-state, cross-word context-dependent Hidden Markov Model (HMM). All the acoustic models were trained with broadcast news data collected from 50 hours of Greek television shows. The corpus for language-model training consists of text from Greek daily newspapers and is used as training data for building both back-off and class-based language models. In the acoustic modeling part of the system, discriminative training is performed and condition-specific acoustic models are created. Audio classification is being performed in order to tag clean speech, speech with background noise/music and telephone speech. The overall word transcription error of our evaluation material was 23%.

Acknowledgements

I feel that I am incredibly fortunate for working under the supervision of Professor Vassilios Digalakis for a second time. I would like to express my sincere gratitude to my advisor for his valuable guidance.

I would also like to thank professors Alexandros Potamianos and Athanasios Liavas for participating in the examining committee.

I fell obliged to thank all my colleagues, Ilias, Pavlos, Theo, Niko, Vaso, Mixail and Maria for all the good times in and out of the lab. Finally i want to thank my parents for all their love and support.

Chapter 1

Introduction

Research on Automatic Speech Recognition (ASR) started in late 1940s through early 1950s and has been active with steady advances for more than half a century. Transcribing speech by machines is a difficult task since it contains not only speech variability such as speaker, noise, or speaking style but also necessity of understanding the topic or the context of speech. Researchers have constrained the problem by constraint to make it possible to be solved. For example, speaker-dependent isolated-digit recognition is a quite constrained condition, however it was a reasonable target for early ASR research. The history of ASR researches has an aspect of relaxation of the constraints, as well as an aspect of development of new techniques.

In the 1970s, the Defense Advanced Research Projects Agency (DARPA) initiated a project to develop speech understanding systems [39]. The goals of the project were to develop systems that would accept 1000-word vocabulary continuous speech from many speakers. In the 1990s, DARPA launched a new project to transcribe read newspaper articles from the Wall Street Journal (WSJ) [56]. It was a challenge to LVCSR (Large Vocabulary Continuous Speech Recognition) using statistical language models trained from large amounts of text data. The vocabulary size increased from 5,000 words to 64,000 words and even more. The speech database was recorded from various speakers using various microphones and the project aimed to develop speaker- and microphone-independent techniques.

Following the WSJ project, the target of DARPA's project was changed to broadcast news speech (BN) [55]. Although the transition from read newspaper speech to broadcast news speech may seem to be natural, it included quite a big change from a viewpoint of relaxing the constraint for speech recognition. Speech data before then was collected for ASR research. Some were utterances for spoken dialogue systems and some were read or dictated speech for transcription. Contrarily, broadcast news speech is not aimed for speech recognition research. Thus broadcast news speech was a new challenge and a big step to practical application for speech recognition.

Although it was first thought to be difficult, automatic transcription of broadcast news is a common LVCSR evaluation task today. A typical current LVCSR system is composed of various components as follows:

- Acoustic Feature Extraction

- Audio Segmentation/Classification
- Acoustic Modeling
- Language Modeling
- Search Algorithms
- Adaptation

Each component is related to various technical fields such as digital signal processing, acoustics, phonetics, pattern recognition, statistical modeling and so on. Thus the ASR research today is an interdisciplinary and integrated field.

1.1 Previous Work

1.1.1 Transcription of Training Corpus

There has been a substantial progress in speech technology, with today's state-of-art systems being able to transcribe unrestricted broadcast news. Aside from the substantial progress, there are many challenging issues in this area. Translation of a continuous speech signal into a sequence of words is a difficult task, as continuous speech does not have any natural pauses between words. The conventional method of building a speech recognizer for any language requires a large amount of segmented and labeled speech corpus for training. However, obtaining such a corpus is a labour intensive and time consuming process. Another challenge is to reduce the cost, both in terms of human effort and financial costs, when it is required to adapt a recognition system to a new task or another language. The quality of automatic segmentation and labeling is potentially of great significance for continuous speech recognition systems, as word-error rate greatly depends on the accuracy of segmentation and sub-word unit recognition [22].

Allwood [33] has pointed out that the frequency of words and grammatical constructions in spoken and written language is vastly different. Therefore, speech cannot be viewed as an exact translation of written language into its spoken form. Further, speech is uttered in an environment of different sounds generally termed noise which consists of unwanted information in the speech signal. Channel variability caused by the prevalent noise and use of different microphones makes speech recognition a formidable task.

As speech recognition involves matching of speech signal with an already existing group of models, the lexicon needs to be kept small, in order to reduce the search space. This causes another problem called *out-of-vocabulary*, which occurs when the intended unit is not present in the lexicon. Automatic speech recognition (ASR) should also be capable of handling out-of-vocabulary issues in an efficient manner. Speaker variability, such as variation in speaking style, gender of the speaker and speaking rate, also affects speech recognition performance.

Researchers have tried several ways to automate speech transcription, without compromising the accuracy of models trained from untranscribed data. Some of the commonly used techniques for speech transcription are briefly explained here.

- Ljolje [46], has used an automatic approach to segmentation and labeling of speech when only the orthographic transcription of speech is available. Orthographic transcription is nothing but a verbatim record of what is actually spoken.

- Kemp and Waibel [38], have proposed a method for unsupervised training of the speech recognizer for TV broadcasts. In this procedure, with the transcribed portion of the data, a bootstrap recognizer is built, which is used to generate transcripts of the untranscribed training material. To exclude the erroneous words from these transcripts, a measure of confidence is applied. As a last step, a new recognizer is trained on the remainder of the hypothesized words.

- Jean-Marc [32], has proposed a hybrid approach that combines HMM and ANN for ASR, that initially uses manually segmented and labeled BREF [42] corpus.

- Dilek [21], has proposed a new method for reducing the transcription effort for training the ASR. The number of training examples to be labeled is reduced by automatically processing the unlabeled examples. The most *informative* (incorrectly recognized) ones are then selected with respect to a given cost function and added to the training set.

The basic idea behind all the above mentioned efforts is, to use an existing speech recognizer to transcribe large amounts of untranscribed data, which can further be used to refine the trained models. For a new language, if no such speech recognizer is available, few hours of data is manually transcribed and is used to build a recognizer. This recogniser is then used to increase the amount of training data by transcribing large quantities of untranscribed data. The two basic problems in the above mentioned techniques are (i) if there is a mismatch in the environment or language during transcription, the recognition performance is expected to be very poor, and (ii) if this newly transcribed data is going to be used for further refining the model parameters, convergence of the training process may be very slow and in few cases, it may be impossible.

The immediate alternative to this problem is to manually transcribe part of the new data, which is taken from a different environment, building models using this data and then using these models to transcribe the rest of the data. A few methods for minimizing the required manually transcribed data considered from a different environment for training, are explained below.

- Matthew [49], has proposed a method in which for both segmentation and clustering problems, the symmetric Kullback-Leibler distance is taken as the solution. The symmetric Kullback-Leibler distance is an effective distance metric to facilitate the detection of long-term statistical differences in speech signals. In this, the system is able to detect changes in acoustic conditions and recognize previously observed conditions and this is used to further pool the data.

- In the experiments carried out by Zavaliagkos [50] at BBN Technologies, completely unsupervised acoustic training data from a conversational speech corpus is combined with 3 hours of manually annotated data. It is shown that a lot of

untranscribed data is needed to achieve comparable levels of performance with transcribed data.

- Frank Wessel [15], has proposed an approach in which a low-cost recognizer trained with one hour of manually transcribed speech is used to recognize 72 hours of unrestricted acoustic data. These transcriptions are then used to train an improved recognizer.

- Lamel [41], has shown that the acoustic models can be initialized using as little as 10 minutes of manually annotated data.

- Gunawardana [1], has proposed an unsupervised adaptation of acoustic models to a domain with mismatched acoustic conditions. Estimation of acoustic models from untranscribed acoustic data is done using Expectation Maximization (EM) algorithm.

1.1.2 Acoustic Model Training

The inital training method for acoustic modeling was the Maximum Likelihood Estimation (MLE) method described in later section. But it has been shown that other training techniques can achieve large improvements in performance in comparison to the conventional MLE training criterion:

- Normadin [53], introduced the Maximum Mutual Information training which optimizes the a posteriori probability of the training utterances and hence the class separability

- Chou et al [6] proposed the Minimum Classification Error criterion where an approximation to the error rate on the training data is optimized

- Droppo et al [31] investigate the feasibility of using an acoustic decision tree to directly model the probability of a given observation. Unlike the common phonetic decision tree, which asks questions about phonetic context, an acoustic decision tree asks questions about the vector-valued observations.

- Povey and Woodland [57] introduce the minimum phone error (MPE) and minimum word error (MWE) criteria for the discriminative training of HMM systems. The MPE/MWE criteria are smoothed approximations to the phone or word error rate respectively.

- Finally a common method in all the broadcast news transcription systems are the acoustic adaptation methods due to the large diversity of broadcast news data: Maximum a Posteriori (MAP) adaptation introduced by Gauvain [20] and Maximum Likelihood Linear Regression (MLLR) presented by Leggetter and Woodland [44].

1.1.3 Language Model Techniques

Additionally, research has been performed in the section of language models. Class-based models are applied but also many techniques have been developed through the years for language model adaptation in order to achieve better performance:

- Digalakis and Oikonomidis [54] have applied a class-based language model leading to an interpolated one, leading to a slight but signifigant improvement in the word error rate.
- Chen and Gauvain [43] present an unsupervised adaptation method for language modeling based on information retrieval techniques.
- Daniel Gildea and Thomas Hofmann [7] propose a novel statistical language model to capture topic-related long-range dependencies.
- Yik-Cheung Tam and Tanja Schultz [73] integrate the Latent Dirichlet Allocation (LDA) approach, a latent semantic analysis model, into unsupervised language model adaptation framework leading to large improvements on the performance of their system.
- Kristie Seymore and Ronald Rosenfeld [40] develop a language model adaptation scheme that takes a piece of text, chooses the most similar topic clusters from a set of over 5000 elemental topics, and uses topic specific language models built from the topic clusters to rescore N-best lists.

1.1.4 State-of-the-art

Current state-of-the-art broadcast news ASR systems for the English language, which is the most well researched language, have performances for Word Error Rate (WER) less than 13% with 10xRT [52]. There is no work of a solid Greek transcription system with which we could compare any results. Instead we can show the ASR results obtained for the ESTER phase II (French language) campaign [18] which obtained 11.9% of WER overall.

They also obtained around 10% of WER for clean speech (studio or telephone), to be compared with 17.9% in the presence of background music or noise. But this also means that ESTER test data has much more easy (clean) conditions than more difficult ones (more noise). The overview article [18] outlined that in a more detailed analysis of the results, unsurprisingly, systems are sensitive to degraded speech quality and to background noise. The best system which obtained the 11.9% WER uses acoustic models with GMMs trained for estimating triphones. The features used are 12th order Mel-Frequency Cepstral Coefficients (MFCC) and the log energy, with first and second order derivatives, giving a total of 39 coefficients. This feature vector is linearly transformed to better fit the diagonal covariance Gaussians used for acoustic modeling. The acoustic models were trained on about 190 hours of BN training data. For the final decoding pass, the acoustic models include 23K position-dependent triphones with 12k tied states, obtained using a divisive decision tree based clustering algorithm with the 35 phones. Two sets of gender-dependent acoustic models were built for each data

type (wideband and telephone). Decoding is performed in three passes, where each pass generates a word lattice which is expanded with a 4-gram LM. Then the posterior probabilities of the lattice edges are estimated using the forward-backward algorithm and the 4-gram lattice is converted to a confusion network with posterior probabilities by iteratively merging lattice vertices and splitting lattice edges until a linear graph is obtained. The words with the highest posterior in each confusion set are hypothesized. For the first and second passes the lattice rescoring step is done using a standard 4-gram language model, while in the third pass rescoring is done with a neural network model and the part of speech (POS) language model. To speedup the third pass, the search space for each audio segment is restricted to the word graph derived from the lattice generated in the second pass. This results in a decoding speed of about 7.5xRT. Unsupervised acoustic model adaptation is carried out for each speaker between the decoding passes making use of the hypotheses of the previous pass. This is done by means of an constrained MLLR adaptation followed by a unconstrained MLLR. For the regular MLLR adaptation, two regression classes (speech and non-speech) are used in the second pass, whereas a data driven clustering with a variable number of classes is used in the third pass. A real time version of the decoding procedure has also been implemented. For this condition the decoding is reduced to two passes with very tight pruning thresholds (especially for the first pass) and with fast Gaussian computation based on Gaussian short lists. This real time version obtained 16.8% WER in the same ESTER test set.

1.2 Thesis Contribution

Our goal is to provide high quality transcripts for Greek radio or TV newscast without any human intervention. The challenge is two-fold. First, everyday broadcast news contains a variety of acoustic conditions. In addition to the typical anchor speech, there is also music, phone interviews, foreign language, to name a few. An ASR system must be able to effectively deal with all conditions. Second, Greek is a challenging language for LVCSR applications. The main reason is the agglutinative nature of the language, from the same root, a very large number of words can be formed by suffixation.

To our best of knowledge, only two papers have been published concerning a Greek broadcast news system. Riedler and Katsikas [35] developed rather simple and conventional speech recognition system that perfomed 38% WER. Dimitriadis et al [10] developed a a distributed system that stores and retrieves Broadcast News data, but in the speech recognition module they achieved WER of 39%. In our thesis by integrating technologies such as large-vocabulary continuous speech recognition, audio classification, class-based language modeling and discriminative training we develop such a broadcast news system that performs significantly better.

In our system, discriminative training such as Maximum Mutual Information (MMI) is performed for the first time for a Greek broadcast system. But the major contributions of this thesis are two. Firstly, we introduce a new form of class-based language model, which exploits the information that is provided from the suffix of the previous word leading us to more robust estimations of the probabilities of word sequences.

Secondly, in the module of audio classification apart from the usual MFCCs, we experiment with wavelet-based features and finally we incorporate these two features with a fusion technique, leading us to better classification results.

1.3 Organization of this thesis

The rest of this thesis is organized as follows: In Chapter 2 the fundamental approach of large vocabulary continuous speech recognition (LVCSR) is outlined. The broadcast news corpus used in this paper is described in Chapter 3. For the training of the acoustic model, there has to be an adequate number of transcribed acoustic data that will be used in order to provide a sufficiently trained acoustic model. For the language model, we need text corpus extracted from daily newspapers, to make robust estimations for any word sequence we may encounter.

Our approach for audio classification is described in Chapter 4. Our classifier consists of three Gaussian Mixture Models (GMM) in order to identify different acoustic conditions. Additionally, different sets of features are being used and a fusion technique for better performance and smaller classification error is developed. In this chapter we present a description of a supervised audio content recognition system, the training procedure of a GMM, and finally our audio classification approach is described in detail.

In Chapter 5 we introduce some basic material on N-gram statistical language modeling. We show the background theory of both word-based and class-based models. We present the WER, perplexities and the hit rate of unigrams, bigrams and trigrams probabilities for all the language models implemented. Finally we show our approach for a class-based language model that exploits the information of the suffix of the previous word, leading us to better performance.

Finally in Chapter 6, we describe in detail the training process of all our acoustic models. We used the HTK toolkit [74] for creating our context-dependent, tied-state, cross-word HMMs. Our experimental parameters of our acoustic models are presented as well as the adaptation process that we performed in order to improve the performance of our recognizer. Additionally, the noise-reduction procedure of our waveforms is presented and finally all our WER results are shown. The lowest overall word transcription error of our evaluation material was 23%. In the last chapter useful conclusions are extracted and potential future work is presented.

Chapter 2

Speech Recognition

2.1 Introduction

In this chapter, the fundamental approach of large vocabulary continuous speech recognition (LVCSR) is outlined. Typical modern LVCSR is formulated in a statistical manner and its formulation, architecture, and components of the architecture are described below.

2.2 Formulation

The standard approach to LVCSR is to assume a simple probabilistic model of speech production whereby a specified word sequence, W, produces an acoustic observation sequence X, with probability $P(W, X)$. The goal is then to decode the word string, based on the acoustic observation sequence, so that the decoded string has the maximum a posteriori (MAP) probability, i.e.,

$$\hat{W} = \underset{W}{argmax} P(W|X) \tag{2.2.1}$$

Using Bayes Rule, Equation 2.2.1 can be written as

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \tag{2.2.2}$$

Since $P(X)$ is independent of W, the MAP decoding rule of Equation 2.2.1 is

$$\hat{W} = \underset{W}{argmax} P(X|W)P(W) \tag{2.2.3}$$

The first term in Equation 2.2.3, $P(X|W)$, is generally called the acoustic model, as it estimates the probability of a sequence of acoustic observations, conditioned on the word string. The way in which we compute $P(X|W)$, for large vocabulary speech recognition, is to build statistical models for subword speech units, build up word models from these subword speech unit models (using a lexicon to describe the composition of words), and then postulate word sequences and evaluate the acoustic model probabilities via standard concatenation methods.

The second term in Equation 2.2.3, $P(W)$ is generally called the language model, as it describes the probability associated with a postulated sequence of words. Such language models can incorporate both syntactic and semantic constraints of the language and the recognition task. Often, when only syntactic constraints are used, the language model is called a grammar and may be of the form of a formal parser and syntax analyzer, an n-gram word model ($n = 2, 3, ...$), or a word grammar of some type. Generally such language models are represented in a finite state network that can be integrated with the acoustic model in a straight forward manner [59].

2.3 Architecture

A typical speech recognition system consists of the basic components shown in Figure 2.1. Acoustic models include the representation of knowledge about acoustics, phonetics, microphone and environment variability, gender and dialect differences among speakers. Language models refer to a system's knowledge of what word is likely to occur and in what sequence. Lexicon consists of a list of possible words and their pronunciations represented by subword sequences.

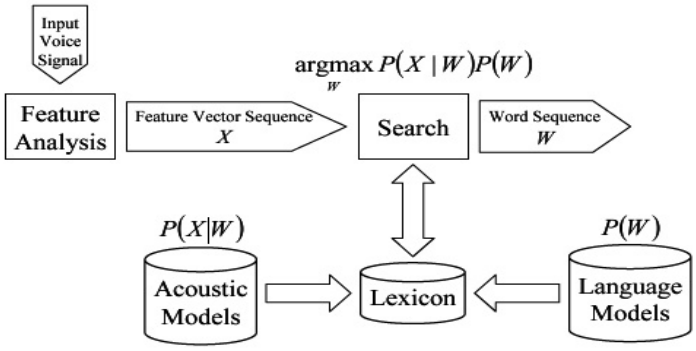

Figure 2.1: Basic system architecture of a speech recognition system

The input voice signal is processed in the feature analysis module that extracts salient feature vectors for the search module. The search module uses both acoustic and language models, which are united via lexicon, to generate the word sequence that has the maximum posterior probability for the input feature vectors.

2.4 Feature Analysis

The role of a feature analysis module is to reduce the data rate, to remove noises, and to extract salient features that are useful for subsequent acoustic matching. The extraction of reliable features is one of the most important issues in speech recognition. In general, time-domain features such as the speech waveform itself are much less accurate than

frequency-domain features such as linear predictive coding (LPC) cepstrum and mel-frequency cepstral coefficients (MFCC). This is because many features such as formants, which are useful in discriminating vowels, are better characterized in the frequency domain with a low-dimension feature vector.

2.4.1 Cepsrtum

The power spectrum of a speech signal is obtained by convolving transfer characteristics of an articulation filter with source signals such as vocal cord vibration. Acoustic characteristics of phonemes mainly depend on modulation transfer characteristics. The cepstrum analysis can decompose power spectrum to the spectral envelope and fine structure. The cepstrum, or cepstral coefficient is defined as the inverse Fourier transform of the short-time logarithmic amplitude spectrum $|X(\omega)|$. $|X(\omega)|$ is given by

$$X(\omega) = G(\omega)H(\omega) \tag{2.4.1}$$

where $G(\omega)$ and $H(\omega)$ are the Fourier transforms of a pseudoperiodic source and vocal tract impulse response, respectively. Its logarithm $\log|X(\omega)|$ is

$$\log|X(\omega)| = \log|G(\omega)| + \log|H(\omega)| \tag{2.4.2}$$

The cepstrum, which is the inverse Fourier transform of $logX(\omega)$ is

$$c(\tau) = F^{\{}-1\}\log|X(\omega)| = F^{-1}\{\log|G(\omega)|\} + F^{-1}\{\log|H(\omega)|\} \tag{2.4.3}$$

where F is the Fourier transform. The first and second terms of Equation 2.4.3 correspond to the spectral fine structure and the spectral envelope, respectively.

When the cepstrum value is calculated by the discrete Fourier transform (DFT), it is necessary to set the base value of the transform, N, large enough to eliminate the aliasing similar to that produced during waveform sampling. The cepstrum then becomes

$$c_n = \frac{1}{N}\sum_{k=0}^{N-1}\log|X(k)|e^{j\frac{2\pi k}{N}n} \tag{2.4.4}$$

2.4.2 LPC Cepstrum

The linear predictive coding (LPC) analysis, which is a very powerful method for speech analysis, models speech signals using an all-pole filter with a sufficient number of poles. The LPC filter is represented as

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.4.5}$$

where p is the order of the LPC analysis.

The above equation means that all-pole spectrum $H(z)$ is used for the spectral density of the speech signal. The LPC spectrum can be obtained from the LPC coefficients

by the following recursion:

$$c_n = \begin{cases} \ln G, & (n = 0) \\ a_n + \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k}, & (0 < n \leq p) \\ \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k} & (n > p), \end{cases} \tag{2.4.6}$$

where G is the gain term in the LPC model.

2.4.3 Mel-Frequency Cepstrum Coefficient (MFCC)

The MFCC is a representation defined as the real cepstrum of a windowed short-time signal derived from the FFT of that signal. The difference from the real cepstrum is that a nonlinear frequency scale is used, which approximates the behavior of the auditory system.

Spectral analysis for MFCC is based on filter bank analysis, where each filter is a triangular filter. The filters compute the average spectrum around each center frequency with increasing bandwidths. The boundary points between adjacent filters are uniformly spaced in the mel-scale. A mel-scale frequency m corresponds to f Hz is calculated by

$$m = 1127.01048 ln \left(1 + \frac{f}{700}\right) \tag{2.4.7}$$

The mel-frequency cepstrum is then the discrete cosine transform of the filter outputs. For speech recognition, typically only the first 13 cepstrum coefficients are used.

2.4.4 Dynamic Features

Temporal changes in the spectra play an important role in human perception. One way to capture this information is to use delta coefficients that measure the change in coefficients over time. Temporal information is particularly complementary to HMMs, since HMMs assume each frame is independent of the past, and these dynamic features broaden the scope of a frame. It is also easy to incorporate new features by augmenting the static feature. The feature vector used for speech recognition is typically a combination of frequency-domain features such as LPC cepstrum or MFCC and their 1st- and 2nd- order dynamic (delta) features.

2.5 Acoustic Modeling

After feature analysis, we have a sequence of feature vectors, X, such as the MFCC vector as our input data. We need to estimate the probability of these acoustic features, given the word or phonetic model, W, so that we can recognize the input data for the correct word. This probability is referred to as acoustic probability, $P(X|W)$.

2.5.1 Hidden Markov Model

The hidden Markov model (HMM) is a very powerful statistical method of characterizing the observed data samples of a discrete-time series. Not only can it provide

an efficient way to build efficient parametric models, but it can also incorporate the dynamic programming principle in its core for a unified pattern segmentation and pattern classification of time-varying data sequences. The data samples in the time series can be discretely or continuously distributed; they can be scalars or vectors. The underlying assumption of the HMM is that the data samples can be well characterized as a parametric random process, and the parameters of the stochastic process can be estimated in a precise and well-defined framework. The HMM has become one of the most powerful statistical methods for modeling speech signals [30].

2.5.2 Context-Dependent Modeling

If there are enough training data to estimate context-dependent parameters, context-dependent units can significantly improve the recognition accuracy. Context-dependent phonemes have been widely used for LVCSR, thanks to its significantly improved accuracy and trainability. A context usually refers to the immediate left and/or right neighboring phones.

A triphone model is a phonetic model that takes into consideration both the left and the right neighboring phones. If two phones have the same identity but different left or right contexts, they are considered different triphones. We call different realizations of a phoneme allophones. Triphones are an example of allophones. A biphone model is also allophone that takes into consideration the left or the right neighboring phones. Biphones can be considered as special cases of triphones that bundle one side of the contexts.

The left and right contexts used in triphones, while important, are only two of many important contributing factors that affect the realization of a phone. Triphone models are powerful because they capture the most important coarticulatory effects. They are generally much more consistent than context-independent phone models. However, as context-dependent models generally have increased parameters, trainability becomes a challenging issue. We need to balance trainability and accuracy with a number of parameter-sharing techniques.

2.6 Language Modeling

Knowledge about language is equally important as acoustic modeling is in automatic speech recognition. For isolated word recognition, lexicon, or list of words to be recognized is linguistic knowledge. Syntactic grammar is designed for spoken dialogue systems. In order to transcribe various expressions of human speech such as broadcast news, it is difficult to design a grammar that covers all possible expressions. Statistical language models which can be trained from text database are flexible enough to cover the variations of human expression. An n-gram language model is a simple and powerful statistical language model, which is broadly used for large vocabulary continuous speech recognition.

2.6.1 N-gram Language Models

As covered above, a language model can be formulated as a probability distribution $P(W)$ over word strings W that reflects how frequently a string W occurs as a sentence. $P(W)$ can be decomposed as

$$P(W) = P(w_1, w_2, ..., w_n)$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_n|w_1, w_2, ..., w_{n-1})$$
$$= \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1}) \quad (2.6.1)$$

where $P(w_i|w_1, w_2, ..., w_{i-1})$ is the probability that w_i will follow, given that the word sequence $w_1, w_2, ..., w_{i-1}$ was represented previously. In Equation 2.6.1 the choice of $w_i$thus depends on the entire past history of the input. For a vocabulary of size v there are v^{i-1} different histories and so, to specify $P(w_i|w_1, w_2, ..., w_{i-1})$ completely, v^i values would have to be estimated.

In reality, the probabilities $P(w_i|w_1, w_2, ..., w_{i-1})$ are impossible to estimate for even moderate values of i, since most histories $w_1, w_2, ..., w_{i-1}$ are unique or have occurred only a few times. A practical solution to the above problems is to assume that $P(w_i|w_1, w_2, ..., w_{i-1})$ depends only on some equivalence classes. The equivalence class can be simply based on the several previous words $w_{i-N+1}, w_{i-N+2}, ..., w_{i-1}$. This leads to an N-gram language model. If the word depends on the previous two words, we have a *trigram*: $P(w_i|w_{i-2}, w_{i-1})$. Similarly, we can have *unigram*: $P(w_i)$, or *bigram*: $P(w_i|w_{i-1})$ language models. The trigram is particularly powerful, as most words have a strong dependence on the previous two words, and it can be estimated reasonably well with an attainable corpus.

2.7 Search

Continuous speech recognition (CSR) is both a pattern recognition and search problem. As described above, the acoustic and language models are built upon a statistical pattern recognition framework. In speech recognition, making a search decision is also referred to as decoding.

The decoding process of a speech recognizer is to find a sequence of words whose corresponding acoustic and language models best match the input signal. Therefore, the process of such a decoding process with trained acoustic and language models is often referred to as just a *search* process. The complexity of a search algorithm is highly correlated with the search space, which is determined by the constraints imposed by the language models.

Speech recognition search is usually done with Viterbi or A* stack decoders. The reason for choosing the Viterbi decoder involve arguments that point to speech as a left-to-right process and to the efficiencies afforded by a time-synchronous process. The reasons for choosing a stack decoder involve its ability to more effectively exploit the A* criteria, which holds out the hope of performing an optimal search as well as the ability to handle huge search spaces. Both algorithms have been successfully

applied to various speech recognition systems. Lately, with the help of efficient pruning techniques, Viterbi beam search has been the preferred method for almost all search recognition tasks. In this study, Viterbi search strategy is employed in the recognition systems.

Chapter 3

Broadcast News Data

An Automatic Speech Recognition system consists of two major parts: the acoustic and the language model. For these two kinds of models we will be more thorough in later chapters. A large amount of data is needed for the training process of both these models.

For the training of the acoustic model, there has to be an adequate number of transcribed acoustic data, that is acoustic signals, that will be used in order to provide a sufficiently trained acoustic model. As for the language model, we need text corpus extracted from daily newspapers so as to make robust estimations for any word sequence we may encounter. In this chapter, we deal with the description of the training data for both the acoustic and the language models.

3.1 Manual Transcriptions

All transcriptions were done using the Transcriber Tool [27], whose output is a textual xml-based version of the transcripts with markup for speaker turns, names, gender, non-speech events and speech condition. The audio transcriptions of our corpus was done in three levels which are described in the next three subsections.

3.1.1 Speech condition level

Transcriber Tool can chop a large waveform into smaller ones. The smaller waveforms that are being produced are tagged with one of the following conditions:

- **report**: clean speech
- **music**: speech with background music
- **noise**: speech with background noise
- **multi speakers**: simultaneous speech from different speakers
- **non greek**: speech other than greek
- **non trans**: no speech

3.1.2 Speaker level

Broadcast news data are very diverse due to different acoustic conditions, different speakers and various channels of speech. During the process of manual transcriptions, we observed the diversity of different speakers. They vary from news speakers, journalists, reporters to citizens interviewed for a report. So the need for speaker tagging was obvious. Table 3.1 sums up the characteristics of each speaker.

Condition	Possible values
Name	*name*
Sex	*male* or *female*
Dialect	*native* or *non native*
Mode	*spontaneous* or *planned*
Channel	*studio* or *telephone*

Table 3.1: Speaker characteristics

Finally, we should establish some ground rules in case where the name of the speaker is not known. So, if the person is not known we give the name *unk_s_n_num* , where *unk* is for unknown user, *s* denotes speaker's sex (either m or f), *n* is his nationality (Greek or non-Greek) and *num* is the number we give to distinguish him from the other speakers.

3.1.3 Transcription level

In this label some basic rules are followed an described in the list below:

1. All sentences must contain Greek words only.
2. Use of lower case letters
3. Numbers are expanded to words
4. Finally table 3.2 shows a number of special markings used during the transcription process.

Event	Symbol
breath	[BREATH]
instant noise	[NOISE]
hesitation	@ε@
misarticulation	**correct articulation**
incomplete words	[FRAGMENT]
bad reading	[TAG_BAD_READING]

Table 3.2: Sound events during speech

3.2 Acoustic Data Corpus

The acoustic training corpus consists of two main data sets, each used for different purpose. First we have the "Logotypografia" [9] corpus which consists of 72 hours of clean speech recorded with two different microphones in a sound proof room. This corpus was split in two sets, one of 21.136 utterances transcribed by the speech recognition group of the Technical University of Crete and one of 10.000 utterances transcribed by the Institute of Language and Speech Processing in Athens.

The "Greek Broadcast" corpus has about 50 hours of various transcribed Greek broadcast news shows recorded from May 2006 to December 2007. These data were obtained from the following shows: ET1, NET, SKAI and MEGA. The videos of these shows have the following characteristics:

- 640x480 resolution
- 1411 kbps bit rate
- 16 bit sample size
- 44 kHz audio sample rate
- PCM audio format
- 25 frames/sec fram rate
- 267 kbps data rate
- 16 bit video sample size
- MPEG-4 video format

After the collection of 50 hours of television shows in video format we wanted to extract the audio information of these videos in audio format. We achieved this extraction with the help of Virtual Dub [29], so as to manipulate the audio data. Finally after the extraction, we led to a total of 50 hours of audio data with the following technical characteristics:

- 256 kbps bit rate
- 16 bit audio sample size
- 1 channel (mono)
- 16 kHz audio sample rate
- PCM audio format

At this point it should be mentioned that only 20 hours of the "Greek Broadcast" corpus is tagged with names, gender etc., while the remaining hours have only their utterance-level transcription. Due to lack of data and incomplete transcribed shows, we decided to split our corpus into only three categories as shown in Table 3.3.

Focus	Description
F0	clean speech
F1	speech with background noise/music
F2	narrow-band speech
F3	other

Table 3.3: Focus Conditions

Finally, the evaluation material was recorded from a different set of dates from the training data and it reflects to various acoustic and channel conditions. The test set that we use in every experiment we perform consists of 933 F0 utterances, 1578 F1 utterances and 102 F2 utterances.

3.3 Language model corpus

In section 1.6 we presented language models from a statistical point of view. In a speech recognition system we could say that the language model is the one that gives prior information about the context of a particular word, or how probable is for that word to be found inside a specific context.

In this thesis we create a recogniser for television broadcast news. We need text data as much relevant with news as possible. The best sources for such a text were Greek online newspapers. Although a newspaper article is written in a more formal style than the way reporters speak, it is the closest corpus to what we need.

Newspaper	Text downloaded (MB)	News period
Elefterotypia	215	1997 - 1999
Ta Nea	170	2002-2009
To Bima	65	2002-2009

Table 3.4: Language Model Corpus

We downloaded text from three different newspapers, the most popular Greek ones: “Eleftherotipia”, “Ta Nea”, “To Vima”. News of political or social interest were selected as primary goals, but we also collected a small amount of sports and financial news. In the first two sources the articles were written in the period 1st of January 2002 to 31st of March 2009. Ta Nea is a daily newspaper not published on Sundays, whereas “To Vima” is a Sunday newspaper. “Eleftherotipia” is covering the period 1997- 1999 and had been collected for the Logotypographia project [9]. In total, we managed to create a corpus of 450 MB.

To format the text, we used a few Perl scripts with a bunch of rules similar to the transcription level. Only Greek letters are used, even for non-Greek words. One sentence per line due to the fact that speech recognizers usually take simple utterances as a unit for processing. After each sentence, a newline character was added. Also we expanded all acronyms. We substituted each one with what it stands for. If it is the first letter of a name, we wrote the whole one or erased it if we do not know it. Additionally, all numbers were written in full text, dates included. Of course, words

like millions or point were added if needed. We removed all punctuation marks and finally we made sure all text is written in lower case.

After the application of these rules, the data is ready to train the language model. Each line of the language model corpus file has one sentence. The frequency of word sequences, will determine the prior probabilities of the language model.

Chapter 4

Audio Classification

In the previous section, while describing our acoustic corpus, we stated that 30 hours of data have not been tagged for speech condition, speaker characteristics etc. So the need for some automatic audio classification method was obvious. In our approach, the manually segmented audio files are clustered. Each cluster is assumed to identify speech in a given acoustic condition leading to a classifier that detects clean speech, speech with background noise/music, and telephone speech.

In supervised classification, emphasis is given to parametric classification methods, including Gaussian, Gaussian mixture model (GMM) and hidden Markov model (HMM) classifiers. Besides the wide popularity of Gaussian and GMM classifiers in general supervised pattern recognition, this emphasis can be justified by the importance of HMMs in time series recognition applications, by the close connection between GMMs and HMMs and by the applicability of both GMMs and HMMs to unsupervised segmentation and classification of a time series. In our approach, our classifier consists of three GMMs in order to identify different acoustic conditions. Additionally, different sets of features are being used and a fusion technique for better performance and smaller classification error is developed. In this chapter we present a description of a supervised audio content recognition system, the training procedure of a GMM, and finally our audio classification approach is described in detail.

4.1 Introduction

Digitally stored multimedia material is currently a rapidly growing resource due to the ongoing technological advancement in data storage, communications and computing. In only recent years, it has become widely possible to transfer long audio and video files via the internet with an acceptable delay. The storage capacities of portable multimedia devices and personal computers have been rapidly increasing. The amount of available digital multimedia information is beginning to overwhelm the capacity of humans to manage and organize it. Thus, computerized solutions for automatic organization of the multimedia material are an attractive approach to access the content efficiently. Because of this, information retrieval is an important field of application for automatic audio recognition. Besides audio material, the indexing and retrieval of (audiovisual) video material also benefits from audio recognition.

Non-speech regions to be discarded can consist of many acoustic phenomena such as silence, music, room noise, background noise, or cross-talk. Earlier work regarding the separation of speech and non-speech (noise, music) classes addressed the problem of classifying known homogenous segments as speech or noise/music. Earlier research also focused more on devising and evaluating characteristic features for classification. Sheirer and Slaney [63], investigated the use of low-energy frames, spectral roll-off points, spectral centroid (correlate of zero crossing rate), spectral flux (delta spectral magnitude) and 4 Hz modulation energy (syllabic rate of speech). El-Maleh [12] used line spectral frequencies for SNS discrimination. More recently, in a completely different approach Williams and Ellis [72] used posterior probability based features for this purpose and Zibert [76] used phoneme recognition based features. Generally in more recent works, cepstral coefficients like PLP [24] and MFCC [51] are used for classifying speech and non-speech signals rather than using specially devised features such as those mentioned for early systems. In fact these Cepstral coefficient features are also widely used in the other types of audio pre-processing classification like gender, background and speaker clustering. Nevertheless they are smoothed to hide or remove aspects of the signal that are not phonetically relevant, such as speaker identity and background noise so they might not be the best basis for distinguishing between different speakers or backgrounds.

The general approach used in classification systems is maximum-likelihood classification with Gaussian mixture models (GMMs) trained on labelled training data to distinguish between several classes of audio [71], although different class models can be used, such as multistate HMMs. The simplest system uses just condition-dependent models while others use four speech models for the possible gender/bandwidth combinations. Noise and music are explicitly modelled in [19] which have classes for speech, music, noise, speech + music, and speech + noise, while Sinha [65] uses wideband music + speech, narrowband speech, music and speech. The extra speech + other models are used to help minimize the false rejection of speech occurring in the presence of music or noise, and this data is subsequently reclassified as speech. The classes can also be broken down further, as in [45], which has eight models in total, five for non-speech (music, laughter, breath, lip-smack, and silence) and three for speech (vowels and nasals, fricatives, and obstruents). When operating on unsegmented audio, Viterbi segmentation, (single pass or iterative with optional adaptation) using the models is employed to identify speech regions. If an initial segmentation is already available , each segment is individually classified. Silence can be removed in this early stage using a phone recognizer [65]. For BN audio, detection performance is around 3% classification error rate (CER), typically less than 1% miss (speech in reference but not in the hypothesis) and 1%-2% false alarm (speech in the hypothesis but not in the reference). When the detection phase is run early in a system, or the output is required for further processing such as for transcription, it is more important to minimize speech miss than false alarm rates, since the former are unrecoverable errors in most systems.

4.2 On Pattern Recognition

A pattern is informally defined as any combination of observable qualities (features) that serves either as a model for replication or an exemplar thereof, i.e., is not just an uninteresting random combination of such qualities. What is interesting is determined by the kind of pattern classes or categories considered; every pattern has an associated class of which the pattern is an exemplar.

Pattern recognition can be informally defined to mean the combined act of 1) detecting the presence of patterns in data (pattern segmentation) and 2) identifying the class which each detected pattern represents (pattern classification) 1. In other words, pattern recognition answers two questions: where are there patterns in the set of observed data and which are the class labels of those patterns. Segmentation answers the where question and classification answers the which question.

A distinction is made between supervised and unsupervised pattern recognition. Unsupervised pattern recognition creates the pattern class specifications directly from the data on the fly, identifies single-class segments and assigns them to the generated classes. In supervised pattern recognition, the pattern class specifications are already known and the problem is to segment and classify the observations so as to label each observation with the identity of some such predefined class.

The canonical pattern recognition system consists of three modules: preprocessing of measurement data, extraction of features from the preprocessed data and the recognition of patterns in the multidimensional space spanned by the feature variables (*feature space*) [11][68]. These modules and their roles in audio signal recognition applications are discussed below.

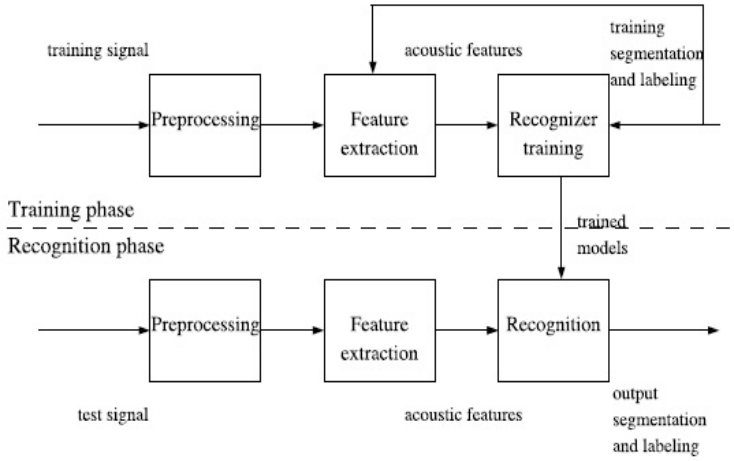

Figure 4.1: A general diagram of a supervised audio content recognition system.

Figure 4.1 shows a diagram of a supervised audio classification/recognition system. The system is first trained in the training phase, shown in the upper half of the diagram.

The actual use of the system is called the recognition phase, shown in the lower half of the diagram. In unsupervised recognition, there is no separate training phase nor labeled training data. Instead, the training and recognition phase are equivalent in unsupervised pattern recognition, as the pattern class descriptions themselves are formed in the same process in which their presence in the data is detected. When training is mentioned below, it should be taken to refer generally to the process used to define the pattern classes, whether in the training phase of supervised pattern recognition or in the course of unsupervised pattern recognition.

Preprocessing is a set of operations performed on a raw digital signal. Preprocessing does not change the essential representation of the data, only some of its qualities. Such operations may include, for example, sampling rate conversion, filtering by a constant *pre-emphasis* filter, as well as *enhancement* operations.

Feature extraction is used to convert the digital signal into a sequence of acoustic *feature vectors*. This requires the feature extraction module to segment the digital signal in time into successive blocks such that each segment block can be associated with a single pattern class (while any true class segment can be represented by more than one feature vector; this is called *oversegmentation*). Feature vectors are typically computed from signal blocks of *constant length* short enough to make the single class assumption valid; the feature vectors are sampled using a constant interval (often in the order of milliseconds). In this case, oversegmentation is drastic and the task of segmentation is essentially left to the recognition module.

Feature extraction may also begin by explicit segmentation of the signal into presumable single-class segments, after which one *segmental* feature vector is produced to describe each segment. These segmental feature vectors are sampled non-uniformly in time. In this case, the recognition module can concentrate more on pattern classification, although some oversegmentation may still be present and additional *smoothing* of the class decisions may have to be performed in order to eliminate suspiciously short (and potentially erroneously labeled) class segments.

Feature extraction has the important task of reducing the dimensionality of the data in order to avoid the effects of the curse of dimensionality [11] which is an ubiquitous problem in pattern recognition. The curse of dimensionality refers to the fact that the more high dimensional is the feature space in which the recognition module must try to model pattern class boundaries, the more training data is required. The amount of required training data grows linearly as a function of unit volume in the feature space and thus exponentially as a function of the number of features [68]. If the curse of dimensionality would not have to be dealt with, feature extraction might not be necessary as patterns could be identified directly from the (preprocessed) digital signals.

Feature selection is an important issue which must be addressed in designing the feature extraction module. It refers to deciding which features to include in the feature vector representation. Often, the features are selected using a combination of domain knowledge and experimentation. Objective methods for feature selection exist and can be applied to a basic set of features selected by the domain expert [23]. Because of the curse of dimensionality, it is important to limit the number of features to those that help in the classification. The way to accomplish this is not to simply aim for

minimal correlation among the features, however, because in doing so important class-discriminatory information may be lost. While it is true that in a pair of two features with a correlation coefficient of unity, the other feature is completely redundant, it is not true that high correlation between two features automatically renders one of them useless. Very high variable correlation or anti-correlation does not mean absence of variable complementarity in terms of pattern class discrimination. In fact, since in reality it is hard to come by features that are truly orthogonal and at the same time helpful in separating the desired pattern classes, it is actually likely that a feature that is highly correlated with another feature having good class discrimination power is itself also a good discriminant between the same classes - maybe offering some additional useful information. Moreover, a variable that is completely useless by itself can provide a significant performance improvement when taken with others [23]. This could not happen if the variables were required to be completely orthogonal.

When the patterns belonging to different classes form distinct, well-separated *clusters* in the feature space defined by the selected features, pattern recognition is easy. For example, if the classes form a clear clustering in a one-dimensional space spanned by a single feature, classification reduces to simple thresholding of the feature value. On the other hand, if the classes are badly overlapping in the feature space, it is often very hard to come by with a classification solution that will separate them. Firstly, this highlights the importance of generating a sufficient number of good features. The class discriminatory information that is lost in the feature representation must be substituted by prior knowledge in the pattern classification stage, if the classes are to be separated. Secondly, if a clustering structure is desired in order to make pattern recognition easier, the features should be carefully selected because including irrelevant features is likely to destroy simple clustering structures that could have been found with a more compact set of good features [37]. All the discussed factors make feature selection necessarily a compromise between information preservation and parsimony.

Pattern classification could be conceptually viewed as regression of the feature vectors on a binary vector with as many elements as there are classes in the classification problem. Each element in this binary vector is 1 if the feature vector pattern belongs to the corresponding class and 0 otherwise. In fact, many classifier structures do contain separate regression modules for each class that regress the feature vectors on a continuous-valued variable (which can be probabilities, for example). The outputs of these regressors are then compared to achieve the class decision (corresponding to the binary vector). On the other hand, the continuous-valued regression outputs can be viewed as features themselves. From yet another viewpoint, the binary vector itself could be viewed as a feature vector fed to a completely trivial pattern classifier (which just determines the location of the value 1 in the binary vector), or it could act as a feature vector for a higher-level recognizer dealing with different classes. As Duda et al [11] point out, there are no particular theoretical reasons for making a strict division of processing steps into feature extraction and classification. The purpose of both tasks is to extract information. In complex solutions with several processing stages, it may be hard to draw a single boundary between the feature extraction and classification stages, as this division depends on the perspective.

4.3 Gaussian Mixture Models

Gaussian mixture models (GMMs) are a widely used parametric classification technique that improve over a single Gaussian distribution and can model a broader, more complex range of distributions using a combination of simple components. They can effectively model distributions where the data points originate from separate clusters but membership is unknown. The idea is to fit the data with regions of probability mass on the assumption that high-level data points are clustered into groups and that each cluster follows a Gaussian distribution (or at least this is a reasonable approximation). This is appropriate as it is anticipated that points emanating from a sound source will typically form a cluster in high dimensional space. Once properly trained a GMM should be effective at predicting the likelihood of a new test sound matching the trained data.

For general audio discrimination, temporal structure is not as important as in speech recognition. GMMs are suitable as they model the whole sample as a mass of points dismissing temporal order (though it can be beneficial to capture the short-term rate of change by a delta-cepstrum). In the case that a sound type is characterised by long-term temporal ordering (e.g. bird song) it may be beneficial to apply a method such as hidden Markov models [17] to capture the characteristic sequence, though this seems unnecessary for the chosen dataset.

GMMs are commonly used in speech research, providing improved discrimination over single Gaussian distributions in many cases. For example, Reynolds [60] achieves good results on databases of differing quality by employing GMMs to verify a speakers voice pattern, in the same manner as we wish to capture the class of a sound. Additionally, Singer et al. develop an effective language identification system using GMMs and delta-cepstrums [64]. Indeed, mixtures are applicable to more general audio problems such as the music classification tasks by Pye [58], and Berenzweig et al. [2].

Though a sound may not necessarily form distinct clusters in acoustic space, the distribution can always be approximated by using enough mixture components. The benefit of modelling the data with a number of simple components is clear but in each case there is the problem of determining the appropriate order of GMM to fit the data.

4.3.1 Parameters of a Gaussian Mixture

To model an acoustic feature x of dimension D the distribution is described by

$$p(x) = \sum_{i=1}^{K} \pi_i p(x|\theta_i) \tag{4.3.1}$$

where K is the number of mixture components, π_i is the probability that component i contributes to modeling the data and θ_i the parameters of component i. In the case of GMMs, $\theta_i = (\mu_i, \Sigma_i)$ and the *probability density function* (pdf) is a multivariate Gaussian function

$$p(x|\theta_i) = \frac{1}{\sqrt{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}}} \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right) \tag{4.3.2}$$

where μ_i is the $D \times 1$ mean vector and Σ_i the $D \times D$ covariance matrix of component i. Hence, for each training class there are two sets of parameters to estimate, the mixing weights π_i and the parameters θ_i. An appropriate value for K must also be determined.

4.3.2 Estimation of parameters

There is no closed form to calculate GMM parameters directly but a maximum likelihood estimate can be obtained over a number of iterative steps, often achieved via the *expectation-maximization* (EM) algorithm. Given a set of training vectors, $x_1, x_2, ..., x_p$ the EM algorithm [8] iteratively refines the GMM parameters to increase the likelihood of the estimated model for the observed data. Starting with a parameter initialization the EM algorithm then proceeds over two steps:

- **Expectation Step**: find the data points 'closest' to a mixture component j, where w_{ij} is the probability that x_i belongs to cluster j using a current estimate of parameters, $p(x|\theta_i)$, is computed as in eq. 4.3.2

$$w_{ij} = \frac{\pi_j p(x_i|\theta_j)}{\sum_{k=1}^{K} w_k p(x_i|\theta_k)} \tag{4.3.3}$$

- **Maximization Step**: calculate new weight $\hat{\pi}_j$, mean $\hat{\mu}_j$ and covariance $\hat{\Sigma}_j$ over P data points to each cluster j

$$\hat{\pi}_j = \frac{1}{P} \sum_{i=1}^{P} w_{ij} \tag{4.3.4}$$

$$\hat{\mu}_j = \frac{1}{P\hat{\pi}_j} \sum_{i=1}^{P} w_{ij} x_i \tag{4.3.5}$$

$$\hat{\Sigma}_j = \frac{1}{P\hat{\pi}_j} \sum_{i=1}^{P} w_{ij} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T \tag{4.3.6}$$

Both steps are repeated (with the maximization estimates becoming the parameters at the next stage) for a set number of iterations or until convergence of the complete-data likelihood, $\prod_{i=1}^{P} \sum_{j=1}^{K} \pi_j p(x_i \theta_j)$. The derivation can be found in [3].

In this work, individual clusters are not represented by a full covariance matrix but by diagonal approximations. The approximation causes the cluster axes to be orientated parallel to the axes of the feature space. This is done for reasons of computational efficiency during parameter estimation. Furthermore, studies by Reynolds have shown that any Gaussian mixture with full covariance can be replicated by a larger order mixture with diagonal covariance and can even outperform full covariance models [60].

Unfortunately, the procedure can converge to different solutions depending on the initialisation of parameters. A good initialisation strategy adopted in this study is to randomly assign each mixture component to a subset of data points and set mixing weights π_j to $\frac{1}{K}$. Several iterations of the K-means algorithm (a non-probabilistic

method) are then used to quickly converge to reasonable estimates of the mean parameters and the EM algorithm then proceeds. It was observed that fewer than ten iterations are needed for convergence. A further problem is that covariances will tend to zero as the likelihood tends to infinity, e.g. if the mixture component models a single point or points close together. To prevent this a size constraint is imposed on covariance values.

4.3.3 Classification

Once a set of classes has been trained by obtaining the parameters π_i and θ_i for each class, the models can be used to predict the most likely class membership of an undefined sample. Classification of a test sound, with feature vector $X = [x_1, x_2, ..., x_N]$ is achieved by estimating the likelihood that each model could have generated X. The log likelihood of a model for the sequence of feature vector is calculated.

$$L(X|\lambda) = \frac{1}{N} \sum_{n=1}^{N} \log \sum_{j=1}^{K} \pi_j p(x_n|\theta_j) \tag{4.3.7}$$

where λ represents all the parameters of a GMM, $\lambda = \{\theta_j, \pi_j, j = 1, ..., K\}$. Summing log values makes the assumption that feature vectors of X are independent. The normalization factor $\frac{1}{N}$ is used to normalise for sample duration, omitting this term would result in longer samples obtaining disproportionately low likelihood values. Reynolds argues that this normalisation factor counteracts the underestimation of actual likelihood values due to the incorrect independence assumption. Likelihood values may be combined with a class prior to determine class membership, however this is not appropriate for this study where there is no prior information of what sound will occur. Thus, to classify a sound, the test vector is tested against all models in turn and assigned to the class of the model predicting the highest likelihood.

4.4 Experiments

An overall view of our system is presented in Figure 4.2. At the preprocessing stage, the input signal is normalized before extracting acoustic features, which are used to characterize the signal. In order to train the classification system, we need to have a sufficient amount of training examples from each class to be recognized. In practice, this is usually achieved by assigning 70% of the manually labeled evaluation data into training set and the rest to the test set. In the pattern learning stage, a representative pattern will be determined for the features of the actual class. This can be done with a model that uses statistical information concerning the features in the training data. In the classification stage, the test data is compared to previously estimated models and classification is done by measuring the similarity between the test data and each model, and assigning the unknown observation to the class whose model is most similar to the observation.

It is difficult to determine what particular features allow us to distinguish between different acoustic conditions. It is even more challenging to find a compact numerical representation for a segment of audio that would retain those distinguishing properties,

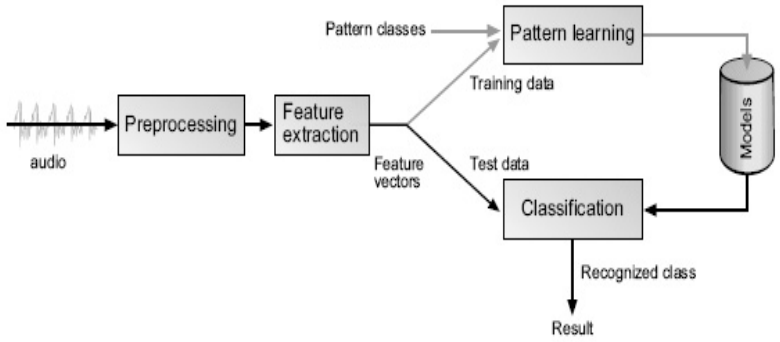

Figure 4.2: Basic system architecture of a speech recognition system

and at the same time discard the irrelevant information. The use of right features is essential for the classification process. There are a wide variety of different features that can be used to characterise audio signals. Features can be divided generally into time-domain and frequency-domain (spectral) features.

Mel-Frequency Cepstral Coefficients have been successfully used in many audio classification problems [13] and they proved to be a good choice for our application, as well. Wavelet-based features were also used and tested, leading us to some useful conclusions described in subsection 4.4.2.

All GMMs that were implemented, were trained on about 1 hour of acoustic data, extracted from the training data after segmentation with the transcriptions. The test set for evaluating our audio classifier consists of 108 utterances of clean speech, 157 utterances of speech with background music/noise and 41 utterances of telephone speech.

4.4.1 MFCC classifier

Mel-frequency cepstral coefficients

Mel-Frequency Cepstral Coefficients (MFCC) is the most widely-used feature in speech recognition [59]. They give a good discriminative performance with reasonable noise robustness. MFCC is a short-term spectrum-based feature, which represents the amplitude spectrum in a compact form. Figure 4.3 shows the steps of extracting the MFCC features. These steps are motivated by perceptual and computational considerations.

The preprocessing step involves pre-emphasising the audio signal, dividing the signal into frames and windowing it. Pre-emphasis is done using a first-order finite impulse response (FIR) filter $1 - 0.97z^{-1}$ to increase the relative energy of high-frequency spectrum. The aim of frame blocking is to segment the signal into statistically stationary blocks. Hamming window is used to weight the pre-emphasised frames. Next, the Discrete Fourier transform (DFT) is calculated for the frames. Since the human auditory system does not perceive pitch linearly, a perceptually meaningful frequency resolution is obtained by averaging the magnitude spectral components over Mel-spaced bins. This is done by using a filterbank consisting of 40 triangular filters occupying the band from

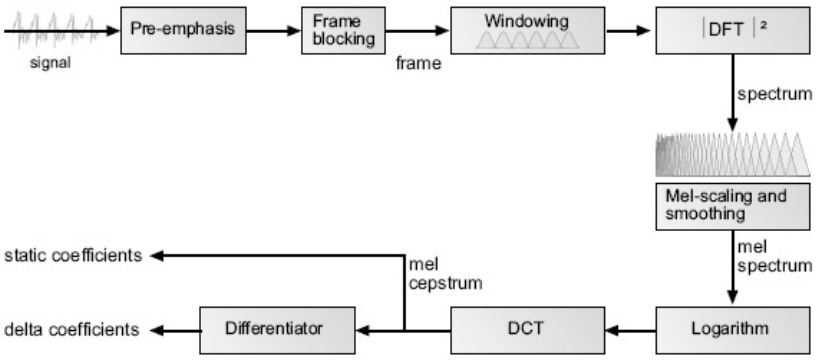

Figure 4.3: Overview of the MFCC feature extraction system

80 Hz to half the sampling rate, spaced uniformly on the Mel-scale. An approximation between a frequency value f in Hertz and in Mel is defined by [25]:

$$Mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{4.4.1}$$

After the Mel-scale filterbank, logarithm is applied to the amplitude spectrum, since the perceived loudness of a signal has been found to be approximately logarithmic. The Mel-spectral components are highly correlated. This is an unwanted property especially for features to be used with Gaussian mixture models, since it increases the number of parameters required to model the features. The decorrelation of the Mel-spectral components allows the use of diagonal covariance matrices in the subsequent statistical modeling. The Mel-spectral components are decorrelated with the DCT, which has been empirically found to approximate the Karhunen-Loeve transform, or, equivalently, the Principal Component Analysis in case of music signals [47]. The DCT is calculated by

$$c_{mel}(i) = \sum_{j=1}^{M} (\log S_j) \cos\left(\frac{\pi i}{M}\left(j - \frac{1}{2}\right)\right) \tag{4.4.2}$$

where $c_{mel}(i)$ is the i^{th} MFCC, M is the total number (40) of channels in filterbank, S_j is the magnitude response of the j^{th} filterbank channel, and N is the total number of the coefficients. For each frame, 17 cepstral coefficients are obtained using this transform and the first coefficient is discarded, as it is a function of the channel gain. The final number of cepstral coefficients is 16, which was found in preliminary evaluations to give sufficient representation of the amplitude spectrum.

Transitions in music carry relevant information and consecutive feature vectors correlate, thus it is important to consider time domain dynamics in feature representation. In addition to the static coefficients, their differentials are also estimated by using a linear regression over consecutive cepstral coefficients. The first-order time derivatives

are approximated with a three-point first-order polynomial fit as follows

$$\Delta c_{mel}(i,u) = \frac{\sum_{k=-1}^{1} k c_{mel}(i,u+k)}{\sum_{k=-1}^{1} k^2} \tag{4.4.3}$$

where $c_{mel(i,u)}$ denotes the i^{th} cepstral coefficient in time frame u [59].

Classification Results of MFCC classifier

Cepstral features were computed from the speech waveforms by a standard algorithm using a log-linear frequency warping function. Each speech frame is represented by a 39-dimensional feature vector that consists of 12 mel frequency cepstral coefficients, normalised log energy along with first and second order derivatives. We experimented with the number of mixture components along with size of the Hamming window. Below, we present a series of tables demonstrating all our classification results.

- **25ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	95	13	0	12
F1	9	148	0	5,7
F2	0	9	32	21
Total Misclassification error : **10,13 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	96	11	1	11,1
F1	19	110	29	30
F2	0	0	41	0
Total Misclassification error : **19,28 %**				

Table 4.1: Confusion matrix with MFCC features with 25ms Hamming Window.

- **30ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	95	13	0	12
F1	14	143	0	8,9
F2	0	7	34	17
Total Misclassification error : **11,10 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	95	13	0	12
F1	9	147	1	6,4
F2	0	5	36	12,2
Total Misclassification error : **9,15 %**				

Table 4.2: Confusion matrix with MFCC features with 30ms Hamming Window.

- **40ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	97	11	0	10,18
F1	13	144	0	8,2
F2	0	10	31	24,4
Total Misclassification error : **11,10 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	94	14	0	12,96
F1	8	148	1	5,73
F2	0	4	37	9,75
Total Misclassification error : **8,82 %**				

Table 4.3: Confusion matrix with MFCC features with 40ms Hamming Window.

- **50ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	93	15	0	13,8
F1	12	145	0	7,64
F2	0	8	33	19,5
Total Misclassification error : **11,40 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	95	13	0	12
F1	15	141	1	10,2
F2	0	5	36	12,2
Total Misclassification error : **11,10 %**				

Table 4.4: Confusion matrix with MFCC features with 50ms Hamming Window.

By careful observation of the above results, we can generally state that the MFCC features perform quite well as we expected from previous works. Classification errors vary from 8% to 20%. The best performance of our system is achieved when 128 mixtures components are trained, using as features MFCC's computed on a 40ms Hamming window with a 10 ms shift. We observe that noisy utterances are classified correctly with a small classification error, while some clean utterances are confused as noisy. Given the diversity of broadcast news data, and the complexity of this task we can say that the performance of the MFCC classifier is satisfying.

4.4.2 Wavelet-based Classifier

Wavelet Audio Analysis

During the past years, wavelets technique have become a common tool in digital signal processing. This kind of analysis has been used in many different research areas including denoising of signals and applications in geophysics (tropical conventions, dispersion of ocean waves etc.). One can conclude that this emerging type of signal analysis is adequate to provide strong solutions in many and completely different researching areas.

The Wavelet Transform (WT) [28] is a technique for analyzing signals. Unlike short time Fourier Trasform (STFT) that provides uniform time resolution for all frequencies, wavelets comprise a dynamic windowing technique which can treat with different precision low and high frequency information.

There are broadly two schemes: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT). In CWT, a mother wavelet (Haar, Daubechies, etc.) is chosen, the signal is multiplied by the wavelet function (window), and the transform is computed separately for different segments of the time-domain signal. The window width is changed for every spectral component and then the transform is computed, which is the most significant characteristic of the wavelet tranform (different from the short-time Fourier or DCT).

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a (coarse) *approximation* coefficients and *detail* coefficients [67]. In this sense, the wavelet decomposition of a signal is similar to *subband coding*, where a signal is passed through several band-pass filters to obtain signal components at different bandwidths, for subsequent analysis and processing. DWT employs two set of functions called *scaling functions* and *wavelet functions*, which are associated with low-pass and high-pass filters respectively. The decomposition of the signal into different frequency bands is obtained by successive low-pass and high-pass filtering of the time domain signal and down-sampling the signal after each filtering. In the discrete domain, a filter is represented by a set of numbers, called *filter taps* which represents the impulse response of the filter. The filtering operation corresponds to the convolution of the discrete signal with the filter.

Figure 4.4 shows the DWT coding process for 3 levels, where $h[n]$ and $g[n]$ denote the low-pass and high-pass filters used by DWT.

After the signal of length, say k, is passed through the filters of any stage l, the outputs, the low and high frequency subbands, are signals of the same length k but with only half the frequency band as input. Thus half of the samples in each of the filter outputs can be removed without loss of information, according to *Nyquist criterion*. This is done by downsampling, where every other sample is discarded. The high frequency subband at stage l is termed *level-l detail coefficients*. The process is repeated a given number of times on the low-frequency subband at each level. The low-frequency subband of the last stage are the approximation coefficients. The combination of these coefficients and the detail coefficients of all levels is termed DWT coefficients.

Since the half-band low-pass filter removes the half of the frequencies present in the input, the frequency resolution of its output is twice that of its input. After the

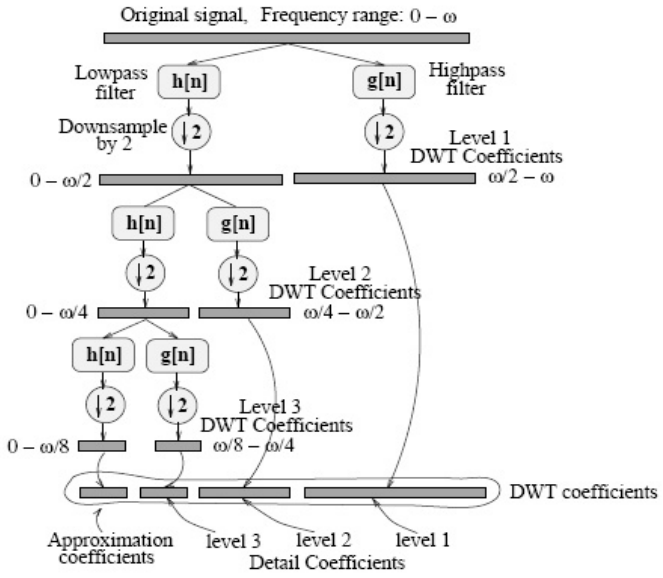

Figure 4.4: The subband decomposition process

downsampling the time resolution is halved since the output contains only half the number of points as the input. Thus, the combination of filtering and downsampling increases the frequency resolution but decreases the time resolution. Only the low-frequency subbands were successively decomposed into two subbands while the high-frequency subbands were untouched. Thus, the approximation coefficients have a high frequency resolution but poor time resolution, while the detail coefficients of level 1 have a poor frequency resolution but good time resolution. Thus, in DWT the time localization of the frequencies are not lost. However, the time localization will have a resolution that depends on which level they appear. Most signals require these kind of resolutions, which makes this scheme attractive for their analysis.

Figure 4.5 shows the original signal, the DWT coefficients and the first 15% of the DWT coefficients for two different audio data. Note that the first 15% of the DWT coefficients carry relevant information. This is the reason for the use of approximation coefficients.

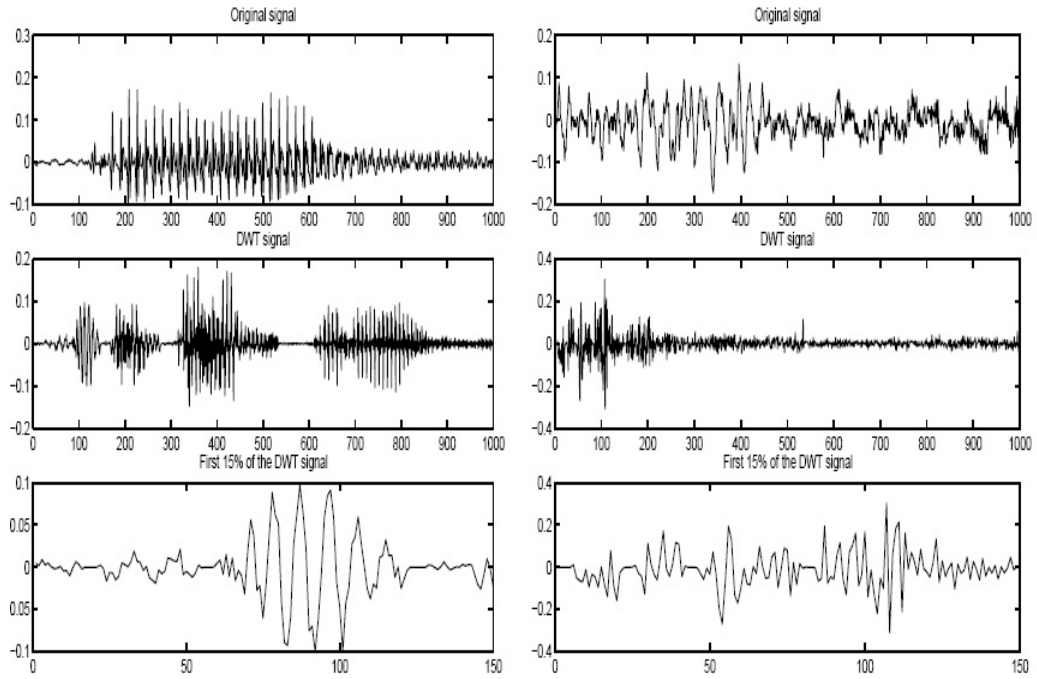

Figure 4.5: Original signal and DWT coefficients of two audio data

Classification Results of Wavelet-based classifier

In our implementation we chose Daubechies 4 as the original (mother) wavelet. DWT is applied in 7 frequency sub-bands of every frame, so a more accurate analysis can be performed, leading to the extraction of detail coefficients for every sub-band. In order to reduce the dimensionality of the extracted feature vectors, statistics over the set of the detail coefficients are computed, leading to 21 dimension features [70]:

- mean: provide information about the frequency distribution of the audio signal
- standard deviation: provide information about the amount of change of the frequency distribution
- median: as mean, provide information about frequency distribution

As with the MFCC classifier, we experimented with the number of mixture components along with size of the Hamming window. We should point out that experiments with different mother wavelets were also performed. We chose not to present them because we observed that in fact different mother wavelets do not affect the performance of the classifier. Below, we present a series of tables demonstrating all our classification results.

- **25ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	97	11	0	10,2
F1	30	125	2	20,4
F2	1	12	28	31,7
Total Misclassification error : **18,3 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	94	14	0	12,3
F1	25	130	2	17,2
F2	1	13	27	34,14
Total Misclassification error : **17,97 %**				

Table 4.5: Confusion matrix with Wavelet-based features with 25ms Hamming Window.

- **30ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	97	10	1	10,18
F1	24	127	6	19,1
F2	0	5	36	12,2
Total Misclassification error : **15 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	94	13	1	12,96
F1	22	132	3	15,92
F2	0	5	36	12,2
Total Misclassification error : **14,37 %**				

Table 4.6: Confusion matrix with Wavelet-based features with 30ms Hamming Window.

- **40ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	96	12	0	11,1
F1	23	131	3	16,5
F2	0	10	31	24,4
Total Misclassification error : **15,6 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	97	11	0	10,18
F1	22	132	3	15,92
F2	0	4	37	9,75
Total Misclassification error : **14,05 %**				

Table 4.7: Confusion matrix with Wavelet-based features with 40ms Hamming Window.

- **50ms Hamming Window**

64 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	93	15	0	13,8
F1	24	130	3	17,2
F2	0	6	35	14,6
Total Misclassification error : **15,68 %**				
128 Mixture Components				
Classified ⟶	F0	F1	F2	error (%)
F0	95	13	0	12
F1	24	130	3	17,2
F2	0	9	32	21,95
Total Misclassification error : **16 %**				

Table 4.8: Confusion matrix with Wavelet-based features with 50ms Hamming Window.

From the results above we can observe that classification error has increased in every case compared to the MFCC classifier. It is clear that MFCC features are more suitable for the acoustic condition task that we are interested in. However if we examine carefully our best Wavelet-based classifier (128 mixture components and 40 ms Hamming window) we can make some useful remarks. First, clean utterances are classified correctly at a higher rate. We have 12,96% misclassification error of the MFCC classifier while the Wavelet-based classifier achieves a misclassification error of 10,18%. Misclassification error for noisy data is increased about 10% compares to MFCC classifier, while the rate of telephone data remains the same.

4.4.3 Fusion of Classifiers

We carefully observed the classification results from table 4.1 to 4.8 and saw that the misclassified utterances for each classifier were different. So we concluded that better results can be achieved if we applied a weight-based fusion technique to combine the MFCC and the Wavelet-based classifiers.
In order to find the weights of the two classifiers we performed 10-fold cross-validation. A random 10% of the audio files were used to test a classifier trained on the remaining 90%. This random partition process was repeated several times and finally extracted an optimal weight α equal to 0.6. Thus the probability of each utterance belonging to a given class is calculated by this equation:

$$p_{fusion} = \alpha * p_{MFCC} + \ (1 - \alpha) * p_{Wavelet} \tag{4.4.4}$$

where p_{MFCC}, $p_{Wavelet}$ are the utterance class probabilities for the MFCC and the Wavelet-based classifiers respectively.

Classified ⟶	F0	F1	F2	error (%)
F0	96	12	0	11,1
F1	8	149	0	5,09
F2	0	4	37	9,75
Total Misclassification error : **7,85%**				

Table 4.9: Confusion matrix of classification with fusion of classifiers.

As we expected, the weight-based fusion technique worked quite well. We get an absolute 1% error reduction for clean utterances and a slight improvement for noisy utterances in comparison to our best MFCC classifier. The classification error rate of telephone data remained untouched.

Chapter 5

Language Modeling

This chapter introduces some basic material on N-gram statistical language modeling. Statistical language modeling tries to acquire regularities of natural language using corpora processing [61]. Corpora are collections of text, e.g., news articles, scientific papers, transcribed dialogues; they can be considered as informative resources. The statistical approach of text processing is naturally being used by researchers due to the categorical character of language and the large vocabularies, in order to estimate numerous parameters [61].

5.1 N-gram Language Modeling

In a n-gram language model, we treat two histories as equivalent if they end in the same $n-1$ words, i.e., we assume that for $k \geq n$, $P(w_k|w_1^{k-1})$ is equal to $P(w_k|w_{k-n+1}^{k-1})$. For a vocabulary size V, a 1-gram model has $V-1$ independent parameters, one for each word minus one for the constraint that all of the probabilities add up to 1. A 2-gram model has $V(V-1)$ independent parameters of the form $P(w_2|w_1)$ and $V-1$ of the form $P(w)$ for a total of V^2-1 independent parameters. In general, an n-gram model has V^n-1 independent parameters: $V^{n-1}(V-1)$ of the form form $P(w_n|w_1^{n-1})$, called the *order-n* parameters, plus the $V^{n-1}-1$ parameters of a (n-1)-gram model.

We estimate the parameters of a n-gram model by examining a sample of text t_1^T, which we call the *training text*, in a process called *training*. If $C(w)$ is the number of times that string w occurs in the string t_1^T, then for a 1-gram language model the maximum likelihood estimate for parameter $P(w)$ is

$$P(w) = \frac{C(w)}{T} \tag{5.1.1}$$

N-gram probabilities are computed by counting and normalizing the N-gram occurrences. For the bigram case the conditional probability of word w_{i-1} given that it is followed by word w_i is computed as

$$P(w_i|w_{i-1} = \frac{C(w_{i-1}w_i}{\sum_w C(w_{i-1}w)}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{5.1.2}$$

Equation 5.1.2 takes the counts of $w_{i-1}w_i$ bigram and divides it by the sum of all bigrams that have w_{i-1} as first word. Note that the latter sum is equal to the count of

w_{i-1} unigram. For the general case of N-gram model the above equation is written as

$$P(w_i|w_{i-N+1}^{i-1}) = \frac{C(w_{i-N+1}^{i-1}w_i)}{C(w_{i-N+1}^{i-1})} \tag{5.1.3}$$

Equations 5.1.2 and 5.1.3 use the frequency interpretation of probability, applying the technique of *Maximum Likelihood Estimation* (MLE). Even with large corpora many N-grams occur only once or they have low counts, so, the computation of N-gram probabilities remains a sparse estimation problem.

5.2 Back-off

Suppose that there are no occurrences of a particular trigram, $w_{i-2}w_{i-1}w_i$ in the training corpus. In this case we can estimate the trigram probability $P(w_i|w_{i-2}w_{i-1})$ using the bigram probability $P(w_i|w_{i-1})$. In the same manner, if there are no counts of the bigram $w_{i-1}w_i$ we can estimate $P(w_i|w_{i-1})$ using the unigram probability $P(w_i)$. This strategy is called *back-off*. According to the above description of backoff method, an amount of probability mass is taken away from the higher-order models and is distributed to the lower-order models [48][34]. Of course, the resulted probability estimation must remain valid, i.e., sums to one.

The backoff model was introduced by Katz [36] and is similar to the deleted interpolation in the sense that the construction of an N-gram model is based on an N-1 model. The difference between backoff and deleted interpolation is that in backoff, for example, if there are non-zero frequency trigram, we use only these counts without interpolating the bigram and unigram models. The back off step downwards to a lower-order model is followed if there are zero counts for the higher-order model.

For a trigram language model, the back-off method is defines as [34]:

$$P(w_i|w_{i-2}w_{i-1}) = \begin{cases} P(w_i|w_{i-2}w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha_1 P(w_i|w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) = 0 \\ & \text{and } C(w_{i-1}w_i) > 0 \\ \alpha_2 P(w_i), & \text{otherwise} \end{cases}$$

Some smoothing techniques assume that the unseen N-grams are all equally probable and an amount of probability mass is distributed it them according to an even scheme. A more neat and fair way is to combine smoothing with backoff for distributing the probability mass to the unseen events. The smoothing quantifies the total mass of probability that must be reserved for the unseen events and the backoff procedure defines how to assign the reserved probability.

The presence of α parameters in the above equation, ensures that the computed probability is a valid probability. This can be explained as follows. If the frequency of the trigram of interest is non-zero, then the $P_{ML}(w_i|w_{i-2}w_{i-1})$ probability that is computed over relative frequencies is a true probability. Otherwise, we have to back off to a lower-order model, and, then, we will add extra probability mass, resulting to a non-true probability. So, the backoff model must be smoothed. Using these

considerations, the $P_{ML}(.)$ must be substituted by smoothed probabilities $P_{SM}(.)$. The use of smoothing saves an amount of probability mass for the lower-order models. Moreover, the α parameters quarantee that the sum of the distributed (to the lower-order models) portions of probability mass is equal to the initially saved amount of probability. In the general N-gram case, the probabilitty mass that must be given form an N-gram to an N-1-gram is defined as follows:

$$\alpha(w_{i-N+1}^{i-1}) = \frac{1 - \sum_{w_i:C(w_{i-N+1}^{i-1})>0} P_{SM}(w_i|w_{i-N+1}^{i-1})}{1 - \sum_{w_i:C(w_{i-N+1}^{i-1})>0} P_{SM}(w_i|w_{i-N+2}^{i-1})} \tag{5.2.1}$$

Note that the α parameter is a function of the history w_{i-N+1}^{i-1}. Also recall that the $P_{SM}(.)$ probabilities are estimated using smoothing. Finally the previous equation is reformulated as:

$$P(w_i|w_{i-2}w_{i-1}) = \begin{cases} P_{SM}(w_i|w_{i-2}w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha(w_{i-2}^{i-1})P_{SM}(w_i|w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) = 0 \\ & \text{and } C(w_{i-1}w_i) > 0 \\ \alpha(w_{i-1})P_{SM}(w_i), & \text{otherwise} \end{cases}$$

5.3 Class-based Language Modeling

Clearly, some words are similar to other words in their meaning and syntactic function. We can benefit from this similarity by grouping words into classes. When there is not enough data to reliably estimate the probability of one word, maybe we can gain some information from the class which that word belongs to. Consider, for example, the bigram *on Thursday*. We may have not seen this bigram in the training data. Instead we have seen the bigram *on Friday*. If *Thursday* and *Friday* belong to the same class, say *days*, we can estimate the probability of *on Thursday* based on the probability of that class.

Let $G : w \longrightarrow g_w$ a class mapping function, mapping each word w of the vocabulary to its word class g_w. Assuming for simplicity that each word belongs to only one class, a class bigram model may have the form [4]:

$$P(w_i|w_{i-1}) = p(g_i|g_{i-1})p(w_i|g_i) \tag{5.3.1}$$

and a class trigram model

$$P(w_i|w_{i-2}, w_{i-1}) = p(g_i|g_{i-2}, g_{i-1})p(w_i|g_i) \tag{5.3.2}$$

That is we first predict the class of the next word based on the classes of the previous words and then, we predict the actual word given its class. Also, these models produce text by first generating a string of classes $g_1, g_2, ..., g_n$ and then converting them into words $w_1, w_2, ..., w_n$ with probability $p(w_i|g_i)$. If we allow words to belong to more than

one class, then we have to sum over all classes that a word belongs to. If $G(w)$ is the set of classes to which the word w can belong then the equivalent equation is

$$P(w_i|w_{i-1}) = \sum_{g_i \epsilon G(w_i)} p(g_i|g_{i-1})p(w_i|g_i) \tag{5.3.3}$$

Class-based models are more compact, that is they have fewer parameters. A n-gram model with vocabulary size V has $V^n - 1$ independent parameters. If we partition the vocabulary into G classes, then the n-gram class model has $G^n - 1 + V + G$ independent parameters. $V - G$ of the form $p(w_i|g_i)$, plus the $G^n - 1$ parameters of an n-gram model for a vocabulary if size G. Also class-based models are more robust, because they share statistics between words of the same class, and are therefore able to generalize to word patterns never encountered in the training data. The drawback of class-based models is that reducing the number of parameters makes the model coarser and thus the prediction of the next word is less precise. So, there has to be a tradeoff between these two extremes.

5.4 Smoothing

The n-gram models are trained from corpora. In practice, every training corpus is of finite size, so, naturally some acceptable n-grams are bound to be absent. This intrinsic characteristic of corpora leads to zero and low counts of n-grams. Using the MLE approach approach the absent n-grams are assigned zero probability, while the probabilities of low-count n-grams are underestimated.

Consider for example the sentence "PETER ATE AN APPLE". If the bigram "PETER ATE" has never occurred in the training corpus, then

$$P(ATE|PETER) = \frac{C(PETER\ ATE)}{\sum_w C(PETER\ w)} = \frac{0}{a}, \qquad a > 0 \tag{5.4.1}$$

Hence, the probability of the whole sentence becomes equal to 0. Clearly, this is an underestimate for the sentence probability, since in real life there is some probability by which the sentence is likely to occur.

Smoothing battles the problem of data sparseness by re-evaluating the zero- and lowprobabilities and assigning them non-zero values. The name of this strategy describes what actually happens. Smoothing techniques make the probability distributions more uniform: adjust low probabilities upward and high probabilities downward [5]. Next, we briefly survey some of the most widely-used smoothing strategies in order to outline the underlying ideas. In this thesis we applied the Modified Kneser-Ney smoothing [5] which is described in the subsection below.

5.4.1 Modified Kneser-Ney smoothing

The modified Kneser-Ney algorithm is an extension of Kneser and Neys algorithm introduced in 1995 which itself is an extension of absolute discounting. Like absolute discounting, the Kneser-Ney algorithm calculates the probability of a word following a particular context by computing the raw probability of the word following the context

and subtracting a discounting amount. This discounting amount is then re-added equally to all n-gram probabilities having the same context, by means of a multiplicative factor that is combined with the probability of the word in the next lower level of the model. That is, the discounted raw probability of the n-gram is linearly interpolated with the smoothed probability of the $(n-1)$-gram created by removing the first word of the context. In absolute discounting, the lower level probability is calculated in the same way as the higher level. However, in Kneser-Ney smoothing, the lower level probability is a smoothed probability calculated not by computing the raw probability of the word following the context, but by computing the number of different contexts that the word follows in the lower order model. The modified Kneser-Ney algorithm is further extended by using three discounting parameters (that, in the highest order model at least, are based on the number of occurrences of the n-gram) instead of the single parameter used in standard Kneser-Ney smoothing and absolute discounting.

Each order of the model is calculated by interpolating between a raw probability for the n-gram and a smoothed probability for the $(n-1)$- gram. In addition, there are different models for the highest order and lower orders of n. For the highest order model, the equation is as follows:

$$P(w_i|w_{i-n+1}^{i-1}) = \frac{C(w_{i-n+1}^{i}) - D(C(w_{i-n+1}^{i}))}{\sum_w C(w_{i-n+1}^{i})} + \gamma(w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1}) \qquad (5.4.2)$$

where

$$D(C) = \begin{cases} 0 & \text{if } C = 0 \\ D_1 & \text{if } C = 1 \\ D_2 & \text{if } C = 2 \\ D_{3+} & \text{if } C \geq 3 \end{cases}$$

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D_1 N_1(w_{i-n+1}^{i-1}\bullet) + D_2 N_2(w_{i-n+1}^{i-1}\bullet) + D_{3+} N_{3+}(w_{i-n+1}^{i-1}\bullet)}{\sum_w C(w_{i-n+1}^{i})} \qquad (5.4.3)$$

and

$$N_1(w_{i-n+1}^{i-1}\bullet) = |w_i : C(w_{i-n+1}^{i-1}) = 1| \qquad (5.4.4)$$

where n_1 is the number of n-grams that appear exactly once, n_2 is the number of ngrams that appear exactly twice, etc.

5.5 Evaluation Metrics

In our experiments, we use three evaluation metrics in order to measure the performance our language models: perplexity, out-of-vocabulary words rate (OOV%), and word error rate (WER).

5.5.1 Perplexity

Statistics are estimated in order to extract the information content of particular data. An n-gram language model can be viewed as a statistical model trying to predict the

next word given n -1 previous words. This approach is based on information theory, which was constructed by Shannon.

The information content of a random variable X is measured by the concept of entropy [62]. Entropy can be viewed as the average uncertainty about occurrence of an event. It is formulated as follows:

$$H(X) = -E\left[logP(X)\right] \tag{5.5.1}$$

So when the probability distribution X with finite alphabet of size L is $p_o, p_1, ..., p_{L-1}$ then entropy can be written as

$$H(p_0, p_1, ..., p_{L-1}) = -\sum_{i=0}^{L-1} p_i log p_i \tag{5.5.2}$$

The main responsibility of a statistical language model is to estimate the true probability distribution of the language, $P(X)$. If the distribution is mis-estimated then the entropy, in other words the uncertainty about the next event will be higher. This entropy is called the cross entropy, which includes the true uncertainty plus the uncertainty added by the wrong estimation of probability distribution:

$$crossentropy(P; P_{LM}) = -\sum_{X} P(X) log P_{LM}(X) \tag{5.5.3}$$

where $P_{LM}(X)$ is the estimated probability distribution of a particular language model. Since $P(X)$ is unknown, another measure should be utilized to find the cross entropy. That measure is called logprob (LP) and is defined as:

$$LP = lim_{n \longrightarrow \infty} -\frac{1}{n} \sum_{k=1}^{n} log P_{LM}\left(w_k | w_1, ..., w_{k-1}\right) \tag{5.5.4}$$

where $\sum_{k=1}^{n} log P_{LM}(w_k | w_1, ..., w_{k-1})$ is the log probability of the long sequence $w_1, ..., w_n$ estimated by the language model. So instead of the ensemble average, time average is taken to estimate the cross entropy with an assumption that language is stationary and ergodic. Since language is neither stationary nor ergodic the logprob is an approximation to the cross entropy.

The logprob is estimated over an unseen text sequence, which is called the test text. If this text appeared in the training text, its probability would be high; consequently the cross entropy would be estimated lower than it should be.

The cross entropy is always higher then the entropy itself, and the quality of a language model is determined from how much the cross entropy gets closer to the entropy of the language. Entropy is also defined as a lower bound on the number of bits or the number of yes/no questions in order to encode or predict the next piece of information. Therefore the average number of choices can be formulated as

$$PP = 2^{LP} \tag{5.5.5}$$

where the quantity PP is called *perplexity*.

From the recognizers point of view, perplexity gives the average number of choices of the next word to be predicted. When perplexity is high, the recognizers task becomes more difficult because of the large number of choices. This difficulty arises from the language itself (the entropy of the language) and the language model (additional uncertainty when the statistics are mis-estimated).

5.5.2 Out-of-Vocabulary Words

The words that appear in the text but are not found in the vocabulary of the recognizer are called the out-of-vocabulary (OOV) words. Due to the agglutinative nature of Greek the OOV words are usually very large.

5.5.3 Word Error Rate

Word Error Rate is simply the percentage of erroneously recognized words to the number of words to be recognized, as in

$$WER = \frac{S + I + D}{N} \times 100 \tag{5.5.6}$$

where S is the number of substitutions, I the number of insertions, D the number of deletions and N the number of total words to be recognized.

5.6 Experimental Procedure and Results

5.6.1 Lexicon size

Through our experiments with the broadcast news data we tested about 6 different vocabulary sizes to check the WER performance of our system and finally make a decision for the optimal lexicon size. Figure 5.1 shows word error rate versus the number of words used for the vocabulary.

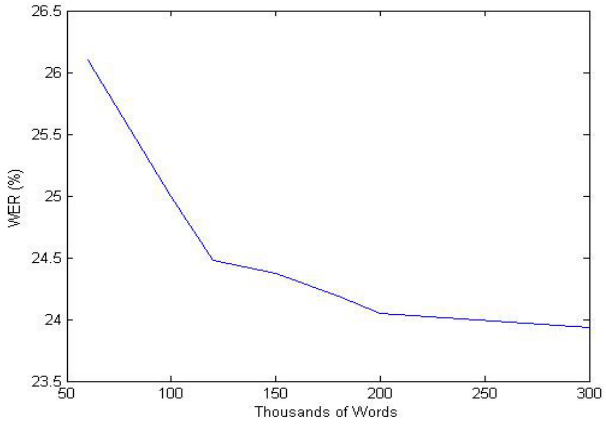

Figure 5.1: WER versus lexicon size

From Figure 5.1 we can clearly observe that with the use of 100K lexicon we have an absolute 1% improvement compared to the 60K lexicon, while we have a similar rate of improvement if we increase the size of our lexicon from 100K to 300K. Thus, we chose to perform all our experiments with the 60K and 100K vocabulary sizes, avoiding computational costs with the use of larger sizes. The same conclusion can be extracted from Figure 5.2, where the OOV rates for different vocabulary sizes are presented. The language models for the particular experiments were obtained from previous work described in [69].

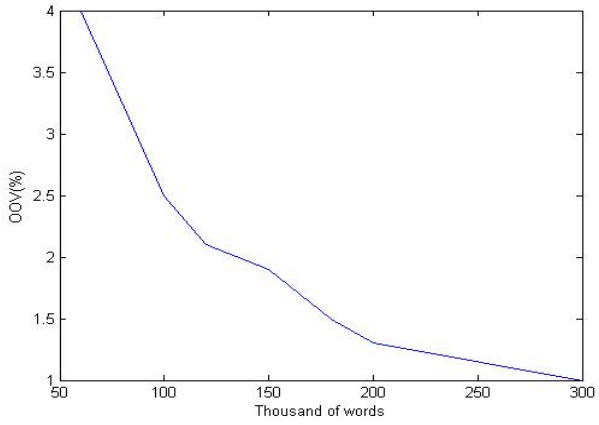

Figure 5.2: OOV versus lexicon size

We can observe that with the use of 100K lexicon, OOV rate reduces 1,5% compared to the 60K lexicon. The exact same rate of reduction we get from 100K to 300K lexicon size.

5.6.2 Back-off models results

We build back-off language models , with the help of SRILM tookit [66], using the Modified Kneser-Ney smoothing method described in section 5.4.1, since this method led us to significant improvements as compared to other smoothing strategies such as Witten-Bell, Good-Turing etc. The training corpus for the training process of the language models is described in detail in Chapter 3 table 3.4. Additionally, both bigram and trigram language models were tested. Details about the acoustic models are mentioned in Chapter 6. Table 5.1 summarizes the results.

60K		100K	
2gram	3gram	2gram	3gram
30.00	27.86	26.78	26.02

Table 5.1: Word Error Rate of back-off models

As expected the best WER results are achieved with the use of trigram language model with a 100K lexicon. Tables 5.2 and 5.3 show the perplexities and the hit rate of 1-, 2-, 3- grams, e.g. the percentage of times the language model used a trigram or backed-off to bigram or unigram probability.

60K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
240.9	10.1	89.9
100K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
262.7	9.7	90.3

Table 5.2: Perplexities and Hit Rate of 1-, 2- , 3- grams of bigram back-off models

60K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
152.1	10.1	29.9	60
100K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
163.8	9.7	30.2	60.1

Table 5.3: Perplexities and Hit Rate of 1-, 2- , 3- grams of trigram back-off models

5.6.3 Class-based models results

As Greek is a morphologically-rich language, we tested the performance of several class-based language models [54]. At first we clustered the 60K vocabulary into 30000 classes, and the 100K classes into about 50000 classes based on the stem of the words with the help of Linguistica toolkit [26]. In Table 5.4 we show some of the classes based on the stem and in Table 5.5 we present a sample of words (w_i) that are part of our lexicon, their stems (g_i) and their suffixes (e_i).

$\varepsilon\pi o\chi$: $\varepsilon\pi o\chi\acute{\alpha}\varsigma, \varepsilon\pi o\chi\acute{\varepsilon}\varsigma, \varepsilon\pi o\chi\acute{\eta}\varsigma, \varepsilon\pi o\chi\acute{\omega}\nu$
$\kappa\lambda\acute{\eta}\rho$: $\kappa\lambda\acute{\eta}\rho o\iota$, $\kappa\lambda\acute{\eta}\rho o\nu$, $\kappa\lambda\acute{\eta}\rho o\varsigma$, $\kappa\lambda\acute{\eta}\rho o\upsilon$, $\kappa\lambda\acute{\eta}\rho o\upsilon\varsigma$, $\kappa\lambda\acute{\eta}\rho\omega\nu$
$\mu\acute{\alpha}\sigma\tau\iota\gamma$: $\mu\acute{\alpha}\sigma\tau\iota\gamma\alpha$, $\mu\acute{\alpha}\sigma\tau\iota\gamma\alpha\varsigma$, $\mu\acute{\alpha}\sigma\tau\iota\gamma\varepsilon\varsigma$
$\nu\upsilon\sigma\tau\acute{\alpha}\zeta$: $\nu\upsilon\sigma\tau\acute{\alpha}\zeta\varepsilon\iota$, $\nu\upsilon\sigma\tau\acute{\alpha}\zeta\omega$, $\nu\upsilon\sigma\tau\acute{\alpha}\zeta o\upsilon\nu$, $\nu\upsilon\sigma\tau\acute{\alpha}\zeta\varepsilon\iota\varsigma$
$\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu$: $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu\alpha$, $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu\varepsilon\varsigma$, $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu\eta$, $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu\eta\varsigma$, $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu o$, $\pi\rho\eta\sigma\mu\acute{\varepsilon}\nu o\varsigma$

Table 5.4: Classes based on stem

Word (w_i)	Stem (g_i)	Suffix(e_i)
$\beta\rho\acute{\varepsilon}\theta\eta\kappa\varepsilon$	$\beta\rho\acute{\varepsilon}\theta\eta\kappa$	ε
$\mu\varepsilon\tau\acute{\alpha}\beta\alpha\sigma\eta$	$\mu\varepsilon\tau\acute{\alpha}\beta\alpha\sigma$	η
$o\lambda\upsilon\mu\pi\iota\alpha\kappa\acute{\alpha}\rho\alpha$	$o\lambda\upsilon\mu\pi\iota\alpha\kappa\acute{\alpha}\rho$	α
$\pi\lambda\alpha\tau\varepsilon\iota\acute{\omega}\nu$	$\pi\lambda\alpha\tau\varepsilon\iota$	$\acute{\omega}\nu$
$\sigma\upsilon\mu\phi\varepsilon\rho o\nu\tau o\lambda o\gamma\acute{\iota}\alpha$	$\sigma\upsilon\mu\phi\varepsilon\rho o\nu\tau o\lambda o\gamma\acute{\iota}$	α
$\upsilon\pi o\theta\eta\kappa\varepsilon\upsilon\mu\acute{\varepsilon}\nu o$	$\upsilon\pi o\theta\eta\kappa\varepsilon\upsilon\mu\acute{\varepsilon}\nu$	o

Table 5.5: A sample of words, their stems and suffixes

After the clustering of words into classes based on their stem we constructed bigram models of the form:

$$p(w_i|w_{i-1}) = p(g_i|g_{i-1})p(w_i|g_i) \tag{5.6.1}$$

and trigram models of the form:

$$p(w_i|w_{i-2}, w_{i-1}) = p(g_i|g_{i-2}, g_{i-1})p(w_i|g_i) \tag{5.6.2}$$

At this point consider the class model based on stem and suppose we want to estimate the probability of the bigram "$\varepsilon\pi\acute{\iota}\mu o\nu\omega\nu\ \pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu$" which we have not seen in the training text. We also assume that we have seen the bigram $\varepsilon\pi\acute{\iota}\mu o\nu\varepsilon\varsigma\ \pi\rho o\sigma\pi\acute{\alpha}\theta\varepsilon\iota\varepsilon\varsigma$. This model uses the probability

$$p(\pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu|\varepsilon\pi\acute{\iota}\mu o\nu\omega\nu) = p(\pi\rho o\sigma\pi\alpha\theta|\varepsilon\pi\acute{\iota}\mu o\nu)p(\pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu|\pi\rho o\sigma\pi\alpha\theta)$$

The probability $p(\pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu|\pi\rho o\sigma\pi\alpha\theta)$ does not any longer depend on the suffix of $\varepsilon\pi\acute{\iota}\mu o\nu\omega\nu$. With this model, the grammatically wrong bigram $\varepsilon\pi\acute{\iota}\mu o\nu\varepsilon\varsigma\ \pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu$ could have as high probability as $\varepsilon\pi\acute{\iota}\mu o\nu\omega\nu\ \pi\rho o\sigma\pi\alpha\theta\varepsilon\iota\acute{\omega}\nu$. To correct this, we interpolate the class model with a word trigram resulting in a model of the following form:

$$p(w_i|w_{i-2}, w_{i-1}) = \lambda p(w_i|w_{i-2}, w_{i-1}) + (1-\lambda)p(g_i|g_{i-2}, g_{i-1})p(w_i|g_i) \tag{5.6.3}$$

where λ the weight given to the word-based model and $\lambda - 1$ the weight given to the class-based model.

In Table 5.6 we show the WER of the interpolated model of eq. 5.6.3. We saw only a slight improvement in the bigram language model with a 60K lexicon, but in all the other cases we saw no improvement. A possible explanation is that the extracted classes by Linguistica are free of errors.

interp_60K		interp_100K	
2gram	3gram	2gram	3gram
29.96	29.00	27.77	26.49

Table 5.6: Word Error Rate of interpolated models

A more interesting measure is the hit rate of the interpolated language model, which is shown in Tables 5.7 and 5.8

interp_60K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
312.1	10.1	89.9
interp_100K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
343.5	9.7	90.3

Table 5.7: Perplexities and Hit Rate of 1-, 2- , 3- grams of bigram interpolated models

interp_60K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
197.9	10.1	29.9	60
interp_100K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
214.5	9.7	30.2	60.1

Table 5.8: Perplexities and Hit Rate of 1-, 2- , 3- grams of trigram interpolated models

We see that we have a major increase on the perplexities of the new interpolated language models. Additionally, using classes based on stem did not change the hit rate for both bigram and trigram models compared to back-off models. Still, we suspected that maybe with the use of a bigger lexicon, which means a bigger number of classes, class based language models would perform slightly better compared to word based language models. Unfortunately, we observed no improvement in WER, perplexities and the hit rates.

Despite the lack of improvement of the interpolated models, we were convinced

that some information must exist in the stem of words. Hence, we followed a different training technique. We used the 30K and 50K classes that were extracted from the 60K and 100K vocabulary, respectively, and expanded the vocabulary by combining the stem of the classes with all valid endings. This process resulted to two new vocabularies of 150K and 320K words, for the 60K and 100K original lexicons, respectively. We keep the original probabilities $p(w_i|w_{i-2}, w_{i-1})$ for the words that were part of the original vocabularies and for the remaining words we used the following equations , which exploits the information that we can extract from the suffix e_{i-1} of the previous word w_{i-1}:

$$p(w_i|w_{i-1}) = p(g_i|g_{i-1})p(w_i|g_i, e_{i-1}) \tag{5.6.4}$$

$$p(w_i|w_{i-2}, w_{i-1}) = p(g_i|g_{i-2}, g_{i-1})p(w_i|g_i, e_{i-1}) \tag{5.6.5}$$

Table 5.9 shows the WER of the expanded language models. We see a slight improvement (average 0.3%) in our test set.

150K		320K	
2gram	3gram	2gram	3gram
29.58	27.51	26.51	25.72

Table 5.9: Word Error Rate of class-based models

The hit rate and the perplexity of the class-based models are shown in Tables 5.10 and 5.11.

150K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
262	10	90
320K		
PP	hit rate (%) 1-gram	hit rate (%) 2-gram
280.5	9.6	90.4

Table 5.10: Perplexities and Hit Rate of 1-, 2- , 3- grams of bigram class-based models

150K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
197	10.1	29.8	60.0
320K			
PP	hit rate (%) 1-gram	hit rate (%) 2-gram	hit rate (%) 3-gram
200.2	9.7	30.1	60.2

Table 5.11: Perplexities and Hit Rate of 1-, 2- , 3- grams of trigram class-based models

We see no improvement on the perplexities of the new language models. Using classes based in stem increased the number of times the model uses a trigram probability only about 1%, despite our expectations that were based on the inflectional nature of the Greek language.

Chapter 6

Acoustic Modeling

In this chapter, we describe in detail the training process of all our acoustic models. We used the HTK toolkit [74] for creating our context-dependent, tied-state, cross-word HMMs. The experimental parameters of our acoustic models are presented as well as the adaptation process that we performed in order to improve the performance of our recognizer. Additionally, the noise-reduction procedure of our waveforms is presented and finally all our WER results are shown. The lowest overall word transcription error of our evaluation material was 23%.

6.1 Experimental Parameters

6.1.1 Acoustic Analysis

The feature extraction procedure is described in subsection 4.4.1. The system's front-end is configured to output 12 mel-scaled cesptral coefficients, the log-energy and their first and second time derivatives for a total 39 values per frame. The cepstral features are computed with a fast Fourier transform (FFT) filterbank and subsequent cepstral-mean normalization on a sentence basis is performed [16].

6.1.2 Selecting Model Units

When using hidden Markov models to model human speech, an essential question is what unit of language to use. Several possibilities exist, such as: words, syllables or phonemes. Each of these possibilities has advantages as well as disadvantages. At a high level, the following criteria need to be considered when choosing an appropriate unit:

- The unit should be *accurate* in representing the acoustic realization in different contexts.
- The unit should be *trainable*. Enough training data should exist to properly estimate unit parameters.
- The unit should be *generalizable*, so that any new word can be derived.

A natural choice to consider is using whole-word models, which have the advantage of capturing the coarticulation effects inherent within these words. When properly trained, word models in small-vocabulary recognition systems yield the best recognition results compared to other units. Word models are both accurate and trainable and there is no need to be generalizable. For large-vocabulary continuous speech recognition, however, whole word models are a poor choice. Given a fixed set of words, there is no obvious way to derive new words, making word models not generalizable. Each word needs to be trained separately and thus a lot of training data is required to properly train each unit. Only if such training data exists, are word models trainable and accurate.

An alternative to using whole-word models is the use of phonemes. European languages, such as English and Greek, typically have between 40 and 50 phonemes. Acoustic models based on phonemes can be trained sufficiently with as little as a few hundred sentences, satisfying the trainability criterion. Phoneme models are by default generalizable as they are the principal units all vocabulary words can be constructed with. Accuracy, however, is more of an issue, as the realization of phonemes is strongly affected by its neighboring phonemes, due to coarticulatory effects.

Phonetic models can be made significantly more accurate by taking context into account, which usually refers to the immediate left and right neighboring phonemes. This leads to biphone and triphone models. A triphone phoneme model takes into consideration both its left and right neighbor phone thus capturing the most important coarticulatory effects. Unfortunately, trainability becomes an issue when using triphone models, as there can be as many as 40x40x40 = 64.000 of them. In practice, most combinations of phonemes do no appear and thus we have a set of 10.000 - 20.000 of triphones to train.

In our speech recognition system we used a set of 28 phonemes, specially designed for the Greek language by a linguist [9]. Table 6.1 shows the set of these phonemes, plus some extra phonemes that represent certain events.

Basic Phonemes	A, v, i, s, o, g, E, l, G, m, J, r, n, t, u, z, x, D, k, T, ly, f, p, d, C, b, N, c
Additional Phonemes	sil (*silence*), hes (*hesitation*), bre (*breath*), sp (*pause*) fra (*incomplete word*), noi (*noise*), tbr (*bad reading*)

Table 6.1: Phonemes of our system

6.1.3 Model Topology

Speech is a non-stationary signal that evolves over time. Each state of an HMM has the ability to capture some stationary segment in a non-stationary speech signal. A left-to-right topology thus seems the natural choice to model the speech signal. Transition from left-to-right enable a natural progression of the evolving signal and self-transition can be used to model speech features belonging to the same state. Figure 6.1 illustrates a typical 3-state HMM common to many speech recognition systems. The first state, the entry-state, and the final state, the exit-state are so called null-states. These states

do not have self loops and do not generate observations. Their purpose is merely to concatenate different models.

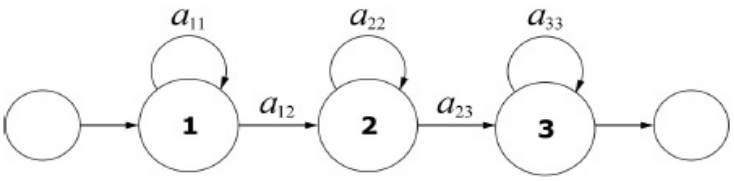

Figure 6.1: Basic structure of a phonetic HMM.

The number of internal states of an HMM can vary depending on the model unit. For HMMs representing a phoneme, three to five states are commonly used. If the HMM represents a word, a significantly larger number of internal states is required. Depending on the pronunciation and duration of the word, this can be 15 to 25 states. More complex transitions between states than the simple topology illustrated in Figure 6.1 are also possible. If skipping states is allowed, the model becomes more flexible, but also harder to train properly.

The choice of output probability function $b_j(x)$ is essential to good recognizer design. Early HMM systems used discrete output probability functions in conjunction with vector quantization. Vector quantization is computationally efficient but introduces quantization noise, limiting the precision that can be obtained. Most modern systems use parametric continuous density output distributions. Multivariate Gaussian mixture density functions, which can approximate any continuous density function, are popular among contemporary recognition systems. A Gaussian mixture density is given as:

$$b_j(x) = \sum_{k=1}^{M} c_{jk} N(x, \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk} b_{jk}(x) \tag{6.1.1}$$

where $N(x, \mu_{jk}, \Sigma_{jk})$, or $b_{jk}(x)$ is a single Gaussian density function with mean vector μ_{jk} and covariance matrix Σ_{jk} for state j, M is the number of mixture components and c_{jk} is the weight of the k^{th} mixture component, which satisfies:

$$\sum_{k=1}^{M} c_{jk} = 1 \tag{6.1.2}$$

6.1.4 Training Criteria for Acoustic Modeling

This section presents a brief review of the training criteria used in acoustic modeling, including *Maximum Likelihood Estimation* (MLE) and *Discriminative Training*. These criteria have been investigated by the speech research community for decades and successfully applied to many real systems. However, they unavoidably have certain weaknesses which become the motivation for new approaches, e.g. ensemble based

methods investigated in this thesis, which aim to achieve better performance than these classic criteria.

Maximum Likelihood Estimation

MLE is the dominant principle that most speech recognition systems are trained with [59] . The performance of MLE training usually works as the baseline for evaluating new training methods. Given the training instances and their class labels, MLE criterion optimizes model parameters by maximizing the class conditional probability of the training data. For speech recognition, this criterion could be expressed as follows. Supposing we have a training set $\Psi(x_i, y_i)|1 \leq i \leq N$ where x_i is the sequence of acoustic features extracted from the i-th utterance in training corpus, and y_i is its transcript, MLE aims to find a model λ^*

$$\lambda^* = \underset{\lambda}{argmax} P(\Psi|\lambda) = \underset{\lambda}{argmax} \prod_{i=1}^{N} P(x_i, y_i|\lambda) = \underset{\lambda}{argmax} \sum_{i=1}^{N} logP(x_i, y_i|\lambda) \quad (6.1.3)$$

Using the chain rule, the probability $P(x_i, y_i|\lambda)$ can be further expressed as

$$P(x_i, y_i|\lambda) = P(y_i|\lambda)P(x_i|y_i, \lambda) \quad (6.1.4)$$

$P(y_i|\lambda)$ denotes the prior probability of the word string y_i being spoken. Its value is provided by the language model. $P(x_i|y_i, \lambda)$ denotes the conditional probability of observing acoustic features x_i given y_i. Its value is provided by the acoustic model.

For most speech recognition systems, the training of language models is a separate process independent of the training of acoustic models. We thus exclude it from the following discussion. Eliminating the influence of language model, the MLE criterion for acoustic modeling can be described as to find an acoustic model λ^* that

$$\lambda^* = \underset{\lambda}{argmax} \sum_{i-1}^{N} logP(x_i|y_i, \lambda) \quad (6.1.5)$$

It has been shown that, if the assumption of the adopted statistical model is correct and sufficient training data is available, MLE training can result in a perfect estimation of class conditional probability $P(x|y, \lambda)$.

However, it's difficult to fulfill these requirements in practice where the amount of training data is limited and the model assumption is usually empirically determined. Moreover, MLE training only considers the correct classes of training instances without taking into account the competing classes. In speech recognition, this may cause unfavorable result. As we know, the transcripts y of an utterance x and other competing hypotheses h that $h \neq y$ often share same words or phonemes. MLE training aiming at maximizing $P(x|y, \lambda)$ could also boost the value of $P(x|h, \lambda)$. As a consequence, the decoding with an acoustic model obtained using MLE may not be able to produce the hypothesis with lowest word error rate.

Discriminative Training

As discussed above, the result of MLE is a set of model parameters which maximize the likelihood of observing the training data given their labeled class. The estimation of the model parameters of a class only depends on the training data belonging to this class. In contrast to MLE, Discriminative Training attempts to incorporate additional knowledge by also using the data associated with competing classes. Namely, discriminative criteria not only try to maximize $P(x|y,\lambda)$, the conditional probability of the training data given the correct class, but also try to minimize $P(x|h,\lambda)$, the conditional probability given alternative classes, and therefore to increase the separability among classes.

The research on discriminative training by the speech community started in the 1980s. Nevertheless, speech community was soon distressed by the observation that the high likelihood doesn't always lead to the low recognition error. Discriminative criteria were then proposed to tackle the shortcomings of MLE training, and evolved to be a family of training approaches with a number of variants. One of them is, *Maximum Mutual Information* (MMI) [53] discussed and practiced in this thesis.

The MMI criterion was proposed as an alternative to MLE in order to overcome the weakness of MLE, in particular the problem of using MLE with an inaccurate model assumption. The idea of the MMI criterion is to minimize the conditional model-based entropy $E_\lambda(h|x)$ of random hypothesis variable h given the input acoustic feature variable x. This corresponds to estimating parameters for model λ that provide as much information as possible about the desired hypothesis given the input pattern x.

The model based entropy for hypothesis variable h is defined as

$$E_\lambda(h) = -\sum_h P_{true}(h) log P(h|\lambda) \tag{6.1.6}$$

where $P_{true}(.)$ denotes the true distribution for data generation while $P(.|\lambda)$ denotes the empirical model based on the estimated distribution. This entropy of hypothesis measures the uncertainty of what hypothesis is spoken without knowing any acoustic information. The conditional entropy of hypothesis given input acoustic feature is defined as

$$E_\lambda(h|x) = -\sum_{h,x} P_{true}(h,x) log P(h|x,\lambda) \tag{6.1.7}$$

This entropy measures the uncertainty to predict what hypothesis is spoken given the input feature sequence x. The amount of information provided by x about h is then represented as the mutual information, the difference between these two entropies.

$$I_\lambda(h;x) = E_\lambda(h) - E_\lambda(h|x) = \sum_{h,x} P_{true}(h,x) log \frac{P(h,x|\lambda)}{P(h|\lambda)P(x|\lambda)} \tag{6.1.8}$$

The goal of MMI estimation is to maximize the mutual information $I_\lambda(h;x)$ by optimizing model λ. As the training data is assumed to be representative, this is equivalent to selecting the value for model λ that maximizes

$$f_{MMI}(\lambda) = \frac{1}{N}\sum_{i=1}^{N} log \frac{P(h=y_i, x=x_i|\lambda)}{P(h=y_i|\lambda)P(x=x_i|\lambda)} = \frac{1}{N}\sum_{i=1}^{N} log \frac{P(x_i|y_i,\lambda)}{\sum_h P(h,x_i|\lambda)} \tag{6.1.9}$$

Comparison between MMI criterion and MLE criterion shows that the latter is only concerned to maximize $P(x|y,\lambda)$, the class dependent conditional probability for the correct label, while the former maximizes the difference between $P(x|y,\lambda)$ and background probability $P(x|\lambda)$. Thus, the MMI criterion is more discriminative than the MLE criterion. The advantage of MMI training is that, maximizing $f_{MMI}(\lambda)$ is more reasonable than maximizing the likelihood of training data when the model assumption is not accurate. However, the computation cost of MMI training is more expensive than that of MLE training due to the need to consider all of the possible classes instead of the correct one only.

6.1.5 Adaptation

In order to make speech recognition systems robust against a continuously changing environment, the use of adaptive techniques is essential. Adaptive techniques are methods to improve the acoustic model accuracy without requiring them to be fully retrained.

Adaptation methods can be either *supervised* or *unsupervised* [75]. In supervised methods the training words or utterances are known to the system in advance, in contrast to unsupervised methods where utterances can be arbitrary. Adaptation methods can be further classified as *on-line* or *off-line*. The on-line methods are used incrementally as the system is in use, working in the background all the time. Off-line adaptation requires a new speaker to input a certain, fixed amount of training utterances. This process is sometimes referred to as enrollment, during which a wide range of parameters can be analyzed. Each of these methods may be appropriate for a particular system, however the most useful approach is on-line instantaneous adaptation. This approach is non-intrusive and generally unsupervised; parameters can be modified continuously while the user is speaking.

Due to the vast diversity of speech, there are three types of adaptation that can be performed:

- speaker adaptation
- enviromental adaptation
- task adaptation

One conventional adaptive technique is *maximum a posteriori* probability (MAP) estimation [20]. This adaptation process is sometimes referred to as Bayesian adaptation. MAP adaptation involves the use of prior knowledge about the model parameter distribution. Hence, if we know what the parameters of the model are likely to be (before observing any adaptation data) using the prior knowledge, we might as well be able to make good use of the limited adaptation data, to obtain a decent MAP estimate. This type of prior is often termed an informative prior. For MAP adaptation purposes, the informative priors that are generally used are the condition independent model parameters. For mathematical tractability conjucate priors are used, which results in a simple adaptation formula. The update formula for a single stream system for state j and mixture component m is:

$$\widehat{\mu}_{jm} = \frac{N_{jm}}{N_{jm}+\tau}\overline{\mu}_{jm} + \frac{\tau}{N_{jm}+\tau}\mu_{jm} \tag{6.1.10}$$

where τ is a weighting of the priori knowledge to the adaptation data and N is the occupation likelihood of the adaptation data, defined as,

$$N_{jm} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jm}^{r}(t) \tag{6.1.11}$$

where R the number of adaptation utterances, T_r the number of frames of r-th utterance, μ_{jm} is the environmental independent mean and $\overline{\mu}_{mj}$ is the mean of the observed adaptation data and is defined as,

$$\overline{\mu}_{mj} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jm}^{r}(t) o_t^r}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jm}^{r}(t)} \tag{6.1.12}$$

where $L_{jm}^{r}(t)$ is the a-posteriori probability of m-th mixture of j state at time t.

As we can see, if the occupation likelihood of a Gaussian component N_{jm} is small, then the mean MAP estimate will remain close to the condition independent component mean. With MAP adaptation, every single mean component in the system is updated with a MAP estimate, based on the prior mean, the weighting and the adaptation data.

6.2 Experimental Results

We used the HTK speech recognition toolkit to train all our acoustic models. After a series of experiments we found that the optimal number of Gaussian distributions for our sustem is 12. Various acoustic models were created with the same procedure: using seed models trained on the "Logotypografia" corpus containing only clean utterances and performing MAP adaptation on our brooadcast news data.

The seed model used is part of the thesis of Dimitris Oikonomidis and is trained on the "Logotypografia" corpus [9]. It consists of 34 monophones, 596 biphones and 6468 triphones. In my bachelor thesis [69] the best acoustic model was constructed by adapting the seed model (**MLS**) trained with MLE on 20 hours of broadcast news data using MAP. The WER results of this model was 35%.

Later, a series of experiments was performed leading to the creation of various acoustic models:

- **MMIS**: Seed model trained with MMI on "Logotypografia" corpus. We used the MLS model as the initial model and to obtain mixture weights.
- **AM1**: We used 30 more hours of additional transcribed broadcast news data, resulting to a 50-hour "Greek Broadcast" corpus MAP adaptation of **MLS**
- **AM2**: 50-hour "Greek Broadcast" corpus MAP adaptation of **MMIS**

In Table 6.2 we present the WER of the MLS, MMIS, AM1 and AM2 models on our evaluation set. All our results use the most efficient language model, the trigram class-based model presented in Section 5. We observe that using MMIS as the seed model, increased the performance by an absolute of 10% in comparison to MLS. The

MLS	MMIS	AM1	AM2
50.96	40.01	25.86	24.92

Table 6.2: Word Error Rate of MLS, MMIS, AM1 and AM2 acoustic models.

AM2 acoustic model has a 1 % gain over the AM1 model due to the use of MMIS as seed model.

Our test set consists of various utterances of F0 (clean speech), F1 (speech with music/noise background) and F3 (telephone speech) conditions. It would be enlightening to see the performance of our system on every single condition. Table 6.3 shows the performance of the AM2 model on different focus conditions.

Focus	AM2
F0	19.32
F1	28.31
F2	53.43
Total	24.92

Table 6.3: Word Error Rate of AM2 on different focus conditions.

From table 6.3 we can see that performance of F1 and F3 utterances is not the one we expected. Especially the F3 utterances have a rather disappointing WER of 53.43%. So in another experiment, we built three different acoustic models to deal with 3 acoustic conditions: clean speech, speech with background noise/music and telephone speech. All 55382 utterances of our 50 hour corpus were classified using the audio classifier of Chapter 4 to three sets that were used to train condition-specific acoustic models. The three training sets consisted of 15825 utterances for F0, 37755 for F1 and 1802 for F3 condition. The training procedure for the condition-specific models (**AM3**) was the same as for AM2. Table 6.4 shows the WER of AM3.

Focus	AM3
F0	19.55
F1	28.45
F2	47.07
Total	24.83

Table 6.4: Word Error Rate of AM3 on different focus conditions.

Still we can see that the performance on telephone data is very poor, due to the small amount of training telephone data, despite the 6% gain.

To improve performance for both telephone and noisy data we performed a Wiener-filter noise reduction over the corresponding waveforms[14]. The rate of noise reduction that is being applied depends on a noise estimate made over the frames judged to be non-speech based on an energy threshold. Additionally, we downsampled our telephone data from 16kHz to 8kHz. After the noise-reduction procedure and the downsampling, we retrained acoustic models (**AM4**) with the same technique. Table 6.5 shows the

total WER as well as the different focus conditions of the condition-dependent acoustic models AM4.

Focus	AM4
F0	19.55
F1	27.60
F2	39.50
Total	23.20

Table 6.5: Word Error Rate of AM4 on different focus conditions.

We observe that there is a significant improvement in the F1 utterances and especially in telephone data, where there is a 8% improvement compared to the AM3 model.

After a series of experiments we managed to elevate the performance of our baseline system (35% WER) to a solid 23% for mixed acoustic conditions, achieving an improvement of 12% absolute. In Figure 6.3 we present the reduction of WER of our system, along with the steps that we followed in order to achieve the final 23% WER. Table 6.6 shows the exact WER of every step.

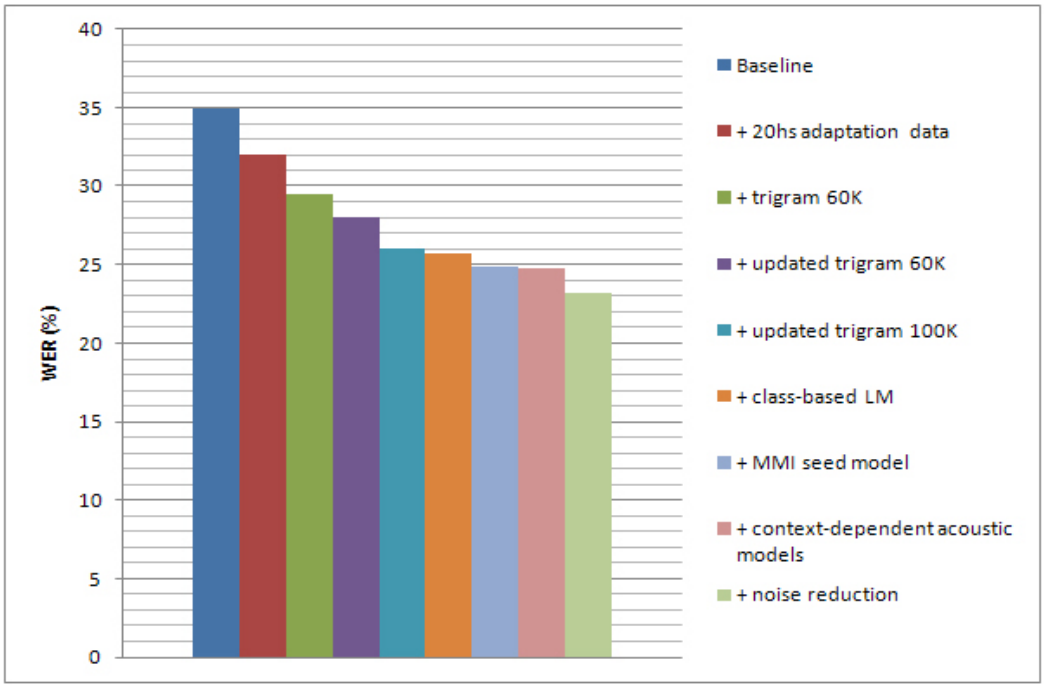

Figure 6.2: Reduction of WER

We can see that conventional methods, like an increase in the amount of adaptation data for the training of acoustic model, the use of a trigram language model instead of a bigram one, the update of the training text and the increase of the lexicon size elevated

	WER (%)
Baseline	35
+ 20 hs adaptation data	32
+ trigram 60K	29.5
+ updated trigram 60K	28
+ updated trigram 100K	26
+ class-based LM	25.7
+ MMI seed model	24.9
+ condition-dependent AM	24.8
+ noise reduction	23.2

Table 6.6: Reduction of WER

the performance of our system with a decrease of 9% in WER. Moreover, class-based language models achieved a slight decrease in the WER. Also, with the implementation of context-dependent acoustic models we did not observed any significant improvement. However using condition-dependent models together with telephone bandwidth and the use of the noise-reduction algorithm improved the WER by a percentage of 2.6%.

Chapter 7

Conclusions and Future Work

We described the corpus and the development process of a Greek broadcast news system. Through a set of different language and acoustic models, we demonstrated a simple and robust recognizer which provided good performance in our evaluation material. Using audio classification, class-based language models, discriminative training and condition-dependent acoustic modeling, we achieved significant improvements on the performance of the system.

There is a number of actions one could take to improve the performance of the system. First of all, what matters a lot is the quality of the data we collect and we use to train our models. Many of the OOV words appeared in several tests, found to be mis-typed words. Hence, a more accurate manual transcription could lead to a well trained acoustic model set.

As far as language modeling is concerned, a 4-gram approach should be tried instead of a trigram LM that we constructed. 4-gram models are the most common and widely used ones in broadcast transcription and they seem to give promising results when applied in other languages. Also a series of experiments could be performed on other language models such as maximum entropy language model etc.

If these techniques are applied and the recognition accuracy level increases, we use to unsupervised training. Instead of manually transcribing more hours to feed the training corpus, we could use the Viterbi decoding for creating transcriptions automatically which could then be added to the training corpus.

Additionally different types of acoustic model adaptation could be applied in our system. The bibliography is full of many other techniques for this kind of adaptation. *Maximum Likelihood Linear Regression* (MLLR) [44], SAT (Speaker Adaptation Techniques) or even a combination of MLLR and MAP [20] could be performed in order to elevate the performance of the acoustic models of the system.

Finally, the most important feature that is absent in our system is a segmentation module. Segmentation aims at finding the changing points between two successive speakers in an audio stream or a change at the speech enviroment. Massive research has been carried out in such techniques. An implementation of a segmentation module would made our system complete due to the fact that we would deal with a fully automatic speech transcriber of broadcast news.

Bibliography

[1] Asela Gunawardana and Alex Acero, "*Adapting Acoustic Models to new domains and conditions using untranscribed data.*" In European Conference on Speech Communication and Technology, 2003.

[2] Berenzweig A., Ellis D.P.W., Lawrence S., "*Anchor space of classification and similarity measurement of music.*" ICME, 2003.

[3] Bilmes J.A., "*A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov Models.*" Technical Report, International Computer Science Institute, Berkeley, California, USA, 1997.

[4] Brown Peter, Desouza Peter, Mercer Robert, Della Pietra Vincent, Lai Jenifer, "*Class-Based N-Gram Models of Natural Language.*" Computational Linguistics, 1990.

[5] Chen F.S., Goodman J., "*An Empirical Study of Smoothing Techniques for Language Modeling.*" In: Proc. Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 1996.

[6] Chou W., Lee C., Juang B., "*Minimum Error Rate Training of inter-word context dependent acoustic model units in Speech Recognition.*" ICALP, 1994.

[7] Daniel Gildeaand Thomas Hofmann, "*Topic-Based Language Models Using EM.*" Eurospeech, 1999.

[8] Dempster A.P., Laird N.M. and Rubin D.B., "*Maximum-likelihood from incomplete data via the EM algorithm.*" Journal of the Royal Society, 1-38, 1977.

[9] Digalakis V., Oikonomidis D., Pratsolis D., Tsourakis N., Vosnidis C., Chatzichrisafis N., Diakoloukas V., "*Large Vocabulary Continuous Speech Recognition in Greek: Corpus and an Automatic Dictation System.*" Interspeech, 2003.

[10] Dimitriadis D., Metallinou A., Konstantinou I., Goumas G., Maragos P., Koziris N., "*Gridnews: a Distributed Automatic Greek Broadcast Transcription system.*" ICASSP, 2009.

[11] Duda R.O., Hart P.E. and Stork D.G., "*Pattern Classification.*" John Wiley and Sons, 2001.

[12] El-Maleh M., Klein G., Kabal P., "*Speech/music discrimination for multimedia application.*" ICASSP, 2000.

[13] Eronen A., "*Comparison of features for musical instrument recognition.*" Technical Report, International Computer Science Institute, Berkeley, California, USA, 1997.

[14] Evans Nicholas, Mason John, "*Computationally Efficient Noise Compensation For Robust Automatic Speech Recognition Assessed under the AURORA 2/3 Framework.*" ICSLP, 2002.

[15] Frank Wessel and Herman Ney, "*Unsupervised training of acoustic models for large vocabulary continuous speech recognition.*" ASRU, 2001.

[16] Fu-Hua Liu, Richard M. Stern, Xuedong Huang, Alejandro Acero, "*Efficient Cepstral Normalization for Robust Speech Recognition.*" ICASSP, 1993.

[17] Furui S., "*Digital Speech Processing, Synthesis and Recognition.*" Marcel Dekker, 2001.

[18] Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K. and Bonastre, J.F. , "*The ESTER Phase II Evaluation Campaign for the Rich Transcription of French Broadcast News.*" Interspeech, 2005.

[19] Gauvain et al., "*Partitioning and Transcription of broadcast news data.*" Academic, 2001.

[20] Gauvain J.L. and Lee C.H., "*Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains.*" IEEE trans on SAP, vol 2, pp.291 - 298, 1994.

[21] Giuseppe Riccardi, Dilek Z. Hakkani-Tur, "*Active and Unsupervised Learning for Automatic Speech Recognition.*" Eurospeech, 2003.

[22] Greenberg S., Chang S. and Hollenback J., "*An introduction to the diagnostic evaluation of Switchboard corpus automatic speech recognition systems.*" NIST Speech Transcription Workshop, 2000.

[23] Guyon I., Elisseeff A., "*An introduction to variable and feature selection.*" Journal of Machine Learning Research, 1157-1182, 2003.

[24] Hermansky et al., "*RASTA-PLP Speech Analysis Technique.*" ICASSP, 1992.

[25] Houtsma A.J.M., "*Hearing, handbook of perception and cognition.*" Academic Press Inc., 1995.

[26] http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/.

[27] http://trans.sourceforge.net.

[28] http://users.rowan.edu/polikar/wavelets/wttutorial.html.

[29] http://www.virtualdub.org.

[30] Huang X., Acero A., Hon H., "*Spoken Language Processing.*" Prentice Hall, 2001.

[31] Jasha Droppo , Michael L. Seltzer , Alex Acero and Yu-Hsiang Bosco Chiu, "*Towards a Non-Parametric Acoustic Model: An Acoustic Decision Tree for Observation Probability Calculation.*" ICALP, 1994.

[32] Jean-Marc and Christophe Ris, "*Development of a speech recognizer using a hybrid HMM/MLP System.*" Europian Symposium on Artificial Neural Networks, 1999.

[33] Jens Allwood E. A.,, "*Corpus-based research on spoken language.*" Nordic Language Technology, 2001.

[34] Jurafsky D., Martin J.H., "*Speech and Language Processing.*" Prentice Hall, 2000.

[35] Jurgen Riedler and Sergios Katsikas, "*Development of a Modern Greek Broadcast-News Corpus and Speech Recognition System.*" Proceedings of the 16th Nordic Conference of Computational Linguistics, 2007.

[36] Katz M.S., "*Estimation of Probabilties from Sparse Data for the Language Model Component of a Speech Recognizer.*" ICASSP, 1987.

[37] Kaufman L. and Rousseeuw P.J, "*Finding Groups in Data: An introduction to Cluster Analysis.*" John Wiley and Sons, 1990.

[38] Kemp T. and Waibel A., "*Unsupervised training of a speech recognizer using TV broadcasts.*" ICASLP, 1998.

[39] Klatt D.H., "*Review of the ARPA Speech Understanding Project.*" Acoustical Society of America Journa;, 62:1345-1366, 1977.

[40] Kristie Seymore and Ronald Rosenfeld, "*Using Story Topics for Language Model Adaptation.*" Eurospeech, 1997.

[41] Lamel, L., Jean-Luc Gauvain and Gilles Adda,, "*Unsupervised acoustic model training.*" ICASSP, 2002.

[42] Lamel L.F., Gauvain J.L. and Eskenazi M., "*BREF, a large vocabulary spoken corpus for French.*" Eurospeech, 1991.

[43] Langzhou Chen, Jean-Luc Gauvain, Lori Lamel, and Gilles Adda, "*Unsupervised Language Model Adaptation for Broadcast News.*" ICASSP, 2003.

[44] Leggetter C.J. and Woodland P.C., "*Maximum Likelihood Linear Regression for speaker adaptation of continuous density HMMs.*" Computer Speech and Language, 1995.

[45] Liu D. and Kubala F., "*Fast Speaker Change Detection for broadcast news transcription and indexing.*" Eurospeech, 1999.

[46] Ljolje A. and Riley M. D., "*Automatic Segmentation and labeling of speech.*" ICASSP, 1991.

[47] Logan B., "*Mel Frequency Cepstral Coefficients for music modeling.*" International Symposium on Music Information Retrieval, 2000.

[48] Manning C.D., Schutze H., "*Foundations of Statistical Natural Language Processing.*" The MIT Press, 2000.

[49] Matthew A. S., Uday Jain, Bhiksha Raj, and Richard M. S, "*Automatic segmentation, classification and clustering of broadcast news audio.*" In Proc. of the DARPA speech recognition workshop, 1997.

[50] Matthew A. S., Uday Jain, Bhiksha Raj, and Richard M. S, "*Utilizing untranscribed training data to improve performance.*" In Proc. of the DARPA Broadcas News Transcription and Understanding Workskop, 1998.

[51] Mermelstein P., "*Pattern Recognition and Artificial Intelligence.*" Academic, 1976.

[52] Nguyen, L., Xiang, B., Afify, M., Abdou, S., Matsoukas, S., Schwartz R., Makhoul J., "*The BBN RT04 English Broadcast News Transcription System.*" Interspeech, 2005.

[53] Normadin Y. and Morgera S.D., "*An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition.*" ICASSP, 1991.

[54] Oikonomidis Dimitrios, Digalakis Vassilios, "*Stem-based Maximum Entropy Language Models for infectional languages.*" Interspeech, 2003.

[55] Pallett D., Fiscus J., Fischer W., Garofolo J., Lund B., Martin A., Przybocki M., "*1994 benchmark tests for the ARPA spoken language program.*" In Proceedings of ARPA Spoken Language Systems Technology Workshop, pages 5-36, 1995.

[56] Paul D. and Baker M., "*The design for the wall street journal-based crs corpus.*" In Proceedings of ICSLP, pages 899-902, 1992.

[57] Povey D., Woodland P.C., "*Minimum phone error and I-smoothing for improved discriminative training.*" ICASSP, 2002.

[58] Pye D., "*Content-based methods for the management of digital music.*" ICASSP, 2000.

[59] Rabiner L., Juang B., "*Fundamentals of Speech Recognition.*" Prentice Hall, 1993.

[60] Reynolds D.A., "*Speaker Identification and Verification using Gaussian Mixture Speaker Models.*" Speech Communication, 17:91-108, 1995.

[61] Rosenfeld R., "*Two Decades of Statistical Language Modeling: where do we go from here?.*" Proceedings of the IEEE, Vol 88, Iss.8, 2000.

[62] Shannon C.E., "*A Mathematical Theory of Communication.*" The Bell System Technical Journal, Vol 27, pp.379-423, 1948.

[63] Sheirer E., Slaney M., "*Construction and Evaluation of a robust Multifeature speech/music discriminator.*" ICASSP, 1997.

[64] Singer E., Kohler M.A., Torres-carrasquillo P.A. and Greene R.J., "*Approaches to language identification using Gaussian Mixture Models.*" ICASSP, 2002.

[65] Sinha R. et al., "*The Cambridge University March 2005 Diarisation System.*" Interspeech, 2005.

[66] Stolcke Andreas, "*SRILM - An Extensible Language Modeling Toolkit .*" in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado, 2002.

[67] Subramanya S.R. and Youssef Abdou, "*Wavelet-based Indexing of Audio Data in Audio/Multimedia Databases.*" in Proceedings of MultiMedia Database Management Systems, 1998.

[68] Theodoridis S. and Koutroumbas K., "*Pattern Recognition.*" Academic Press, 2003.

[69] Tsergoulas Orfeas, Digalakis Vassileios , "*Greek Broadcast Systam.*" Bachelor Thesis, TUC, 2007.

[70] Tzanetakis George, Essl George and Cook Perry, "*Audio Analysis using the Discrete Wavelet Transform.*" in Proc. Conf. in Acoustics and Music Theory Applications, 2001.

[71] Vandecastseye A., Martens J., "*A fast, accurate and stream-based speaker segmentation and clustering algorithm.*" Academic, 2004.

[72] Williams G., Ellis D., "*Speech/music discrimination based on posterior probability features.*" Eurospeech, 1999.

[73] Yik-Cheung Tam and Tanja Schultz, "*Unsupervised Language Model Adaptation Using Latent Semantic Analysis.*" Eurospeech, 2004.

[74] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Olasson, D., Povey, D., Valtchev, V., Woodland, P., "*The HTK Book (for HTK Version 3.2).*" Cambridge University Engineering Department, 2002.

[75] Young S.J., "*Large Vocabulary Continuous Speech Recognition: a Review.*" Cambridge University Engineering Department, 1996.

[76] Zibert J., Mihelic F., Martens J., Meinedo H., Neto J., Docio L., Garcia-Mateo C., "*The COST278 Broadcast News Segmentation and Speaker Clustering Evaluation - overview, methology, systems, results.*" Interspeech, 2005.