NTNU
Norwegian University of
Science and Technology

Evaluating Vowel Pronunciation in Computer Assisted Pronunciation Training.

Stian Erichsen

Master of Science in Communication Technology
Submission date: June 2011
Supervisor: Magne Hallstein Johnsen, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

The ability to speak Norwegian is mandatory for all people with respect to both work and social life. However, acceptable pronunciation quality can be difficult to achieve for persons with another mother language (so called L2-speakers).

Normally L2-speakers will attend courses in Norwegian. However, computer assisted learning is a viable alternative or supplement. This especially applies to early stages where correct pronunciation of the basic sounds are to be learned.

In Norwegian it is especially important to learn to distinguish between short and long versions of vowels; i.e vowel quantity. This is best learned by pronouncing pairs of isolated words which only differ in vowel length (i.e.'hane' versus 'hanne'). However, also vowel quality confusion (i.e. wrong vowel type) is important to minimize.

In this thesis the candidate is to use a CAPT-system for investigating the vowel quantity and quality for a speech database of isolated pairwise words spoken by Norwegians (L1-speakers) and L2-speakers from Iran (farsi) and China (mandarin). The speakers were told which word to pronounce, and thus which vowel to pronounce and whether it should be short or long. The database has been manually annotated at the phoneme level, thus the true vowel pronunciations are known.

The CAPT-system consists of a speech recognizer (ASR) followed by a verifier of the quantity or the quality. The former will find the vowel segments corresponding to the vowel hypotheses and simultaneously do a preliminary decision. The segments are then analyzed and the derived features are used to verify (accept or reject) the ASR-decision.

The candidate shall initially evaluate the performance of the recognizer. Ideally an error free recognizer is required, however, this is not possible to achieve. Further, the errors should be evaluated with respect to recognizer imperfections and pronunciation errors (both of the vowel quantity and quality, and other errors). Finally several features shall be evaluated for use in a verifier. The evaluation criterion should be based on equal error numbers or rate (false rejection and false acceptance).

Assignment given: 15. February 2011

Supervisor: Magne H. Johnsen, IET

Preface

This thesis is the completion of my MSc degree in Communications Technology, and was carried out at the Department of Electronics and Telecommunications at the Norwegian University of Science and Technology (NTNU).

I would like to use this opportunity to thank my supervisor, Professor Magne Hallstein Johnsen, for all the helpful guidance and feedback he has offered me throughout the work on the thesis.

Abstract

Computer Assisted Pronunciation Training (CAPT) applications are tools that can be used when learning a second language. By evaluating the speech of a student, the CAPT system is able to give automatic feedback on his or her pronunciation performance.

Two important properties in Norwegian pronunciation is the quantity and quality of the vowels. It is therefore important that students get to practice this.

Feedback was produced by an ASR based CAPT system, where a speech recognizer evaluated the pronunciation produced by different speakers. However, sind ASR is prone to errors, verification was later performed to test the correctness of the recognizers results.

The recognizer had an error-rate of 7.5 % when evaluating vowel quantity, and an error rate of 42.1 % when evaluating vowel quality. After verification, the first error rate was reduced to 1.35 % by rejecting 7.2 % of the results. The second error rate was reduced to 27,7 % by rejecting 23.5 % of the results.

The use of such a system could therefore be justified for evaluating the vowel quantity in the pronunciation, but not vowel quality.

Contents

1 Introduction

This section will serve as the introduction to the thesis. It will present the research topic, along with a description of the problem to be addressed. First, section 1.1 will discuss the background of the topic, and the reasons for choosing it. Secondly, section 1.2 will present the scope and limitations. Lastly, section 1.3 will give a brief outline of the different sections of the thesis.

1.1 Motivation and Background

Computer assisted language learning (CALL) applications are applications designed to help teachers and students with the language learning process, and can be used either as supplements or even alternatives to traditional language learning. Not only do they free up workload for the teachers, but they also give the students a teacher with unlimited time and patience, that they can use in a stress-free environment when the time suites them.

Computer assisted pronunciation training (CAPT) is a part of CALL, and while CALL tries to incorporate all aspects of language learning, CAPT applications only focus on the pronunciation training. These applications aim to provide automatic and instantaneous feedback on the overall pronunciation quality of a user.

Hansen [8] explains how these feedbacks traditionally have been mainly visual, for example in the form of spectrograms or waveforms. This requires the user to compare the visual content to a model of the target word or sentence, as pronounced by a native speaker or a teacher. He further explains that although this might be helpful for learners of tone language or prosodic training, it does not inform of possible segmental errors. In addition, these systems do not directly inform the user of what is wrong, but instead relies on the users interpretation of the feedback and their ability to self-correct through trial-and-error. To combat this, Hansen [8] suggests that CAPT should make use of Automatic Speech Recognition (ASR), and that feedback should be specific and correspond to the four 'K's: 1) Quantitative 2) Qualitative 3) Comprehensible and 4) Corrective.

Advances in ASR the last two decades has made it possible to use it in CAPT. Studies, [13, 12, 14], on ASR based CAPT applications have shown not only that these applications can be effective, but also that the users have a positive reaction to working with these systems. However, ASR systems have their drawbacks. The error rate, meaning the percentage of erroneous recognitions, can be significantly high, escpecially when recognition is performed on speech from users who are not yet fluent in the language. Errors in the speech recognizer may lead to errors in other parts of the system, and, in turn, give misleading feedback to the user, i.e. rewarding bad pronunciation or punishing good pronunciation.

One comercial CALL application that utilizes ASR based CAPT is Tell Me More French [6], which was evaluated by Reesner in [16]. Figure 1 shows how this application combines the use of visual feedback with feedback based on a score generated using ASR. After attempting a pronunciation, the student is able to listen to the voice recording and compare it to the teacher voice, compare the produced spectrograph to a model, and see the automatically generated score.

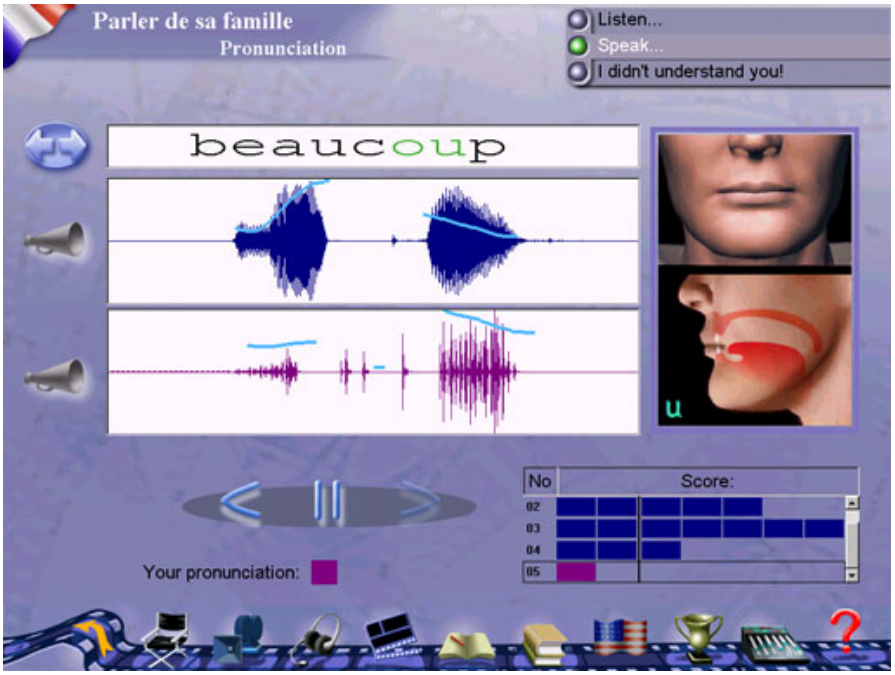

Figure 1: User interface from pronunciation section of Tell Me More French, captured from [16]

No CAPT system currently exists in Norwegian, and it would therefore be beneficial if a system like Tell Me More French could be adapted so that it could be used in another languages, and not just French. However, different languages have such large differences in pronunciation that this is usually not feasible. Therefore, a specialized CAPT system is needed for each language.

1.2 Scope and Limitations

As mentioned in the previous section, one of the problems with using speech recognition in CAPT is that it is prone to errors, and that misleading feedback might be given to the user whenever an error is made. This confusion can reduce the effectiveness of the training.

One way of preventing this is by introducing a verifier, which will try to verify whether the speech recognizers results are correct or not. Then, if the results are verified as correct, proper feedback will be given. If not, the recognizer is assumed to have made an error, and no feedback will be given. The scope of this thesis is to evaluate such a system for Norwegian pronunciation training, where the users are non-natives who are learning Norwegian as a second language.

There are several levels of pronunciation training, ranging from learning how to pronounce different sounds and words, to learning how to use those words in a greater context and carry a dialogue. For simplicity reasons, and to be able to finish the thesis within the given time-frame, the following limitations were set:

- The users are assumed to be in an early stage of the learning process, where they are asked to listen to a word and try to repeat that word.

- The system will only evaluate the pronunciation of the stressed vowel in each word, i.e. the vowel in the stressed syllable, and assume that all the other phonemes are correctly pronounced.

The first limitation makes each pronunciation an isolated-word case, where the job of the speech recognizer is to find the sequence of phonemes that make up the pronounced word. The verifier will then try to verify the correctness of that sequence.

The second limitation takes out-of-vocabulary, insertion and deletion errors out of the equation. The only concern is whether the user pronounced the stressed vowel correctly, or if he/she pronounced some other vowel. The verifier will therefore only have to verify one phoneme, and not the entire sequence of phonemes. The reason for focusing on the stressed vowel is because of its importance in Norwegian pronunciation. If the wrong vowel is pronounced, or if the length of the vowel is incorrect, it can change the meaning of the whole word.

This can be exemplified by the three Norwegian words 'Line', 'lyne', and 'lynne'. The only difference between the pronunciations of these three words is the pronunciation of the stressed vowel. The first word is pronounced with a long pronunciation of the vowel i, while the second and third word are pronounced with a long and short pronunciation of the vowel y, respectively. It is clear that if either the length or the vowel type is incorrect, then one of these words can be mistaken for another.

1.3 Outline of the Thesis

The report is divided into 5 main sections:

Section 1: Introduction.

Section 2: Theory. This section will present the speech recognition and verification system used to evaluate the speech, and the concepts and theoretical background behind it.

Section 3: Methods. This section will explain how the speech recognitin and verification system was investigated. This includes a description of the reseach method used, a description of the speech database and tools that were used, and a description of the methods that were applied to perform verification on both vowel quality and quantity.

Section 4: Results and Discussion. This section will present the results, and discuss the implications of them.

Section 5: Conclusion and Future Work. This section will serve as the closing to the report, summarizing what has been presented and discussed, and what can be concluded from this.

2 Theory

The purpose of this section is to present the necessary theory behind the speech recognition and verification system that is used in this thesis. This theory will then be used for reference later. First section 2.1 will give an overview of the system, and the different components of it. Secondly, section 2.2 and 2.3 will discuss the concept of speech, and different ways that speech can be represented. Finally, section 2.4 and 2.5 will present the basic theory behind the ASR and verification technology that will be used.

2.1 Speech Recognition and Verification System

As mentioned in the introduction, the purpose of this system is to first perform speech recognition on isolated speech and then do a verification on the correctness of the speech recognition results. The reason for doing this is because speech recognition is prone to errors, and therefore, there are two sources of error in the system: The students pronunciation and the speech recognizers classification. To preven erroneous feedback, the verifier will try to reject all incorrect recognitions, and accept all correct recognitions. This is illustrated in 2.

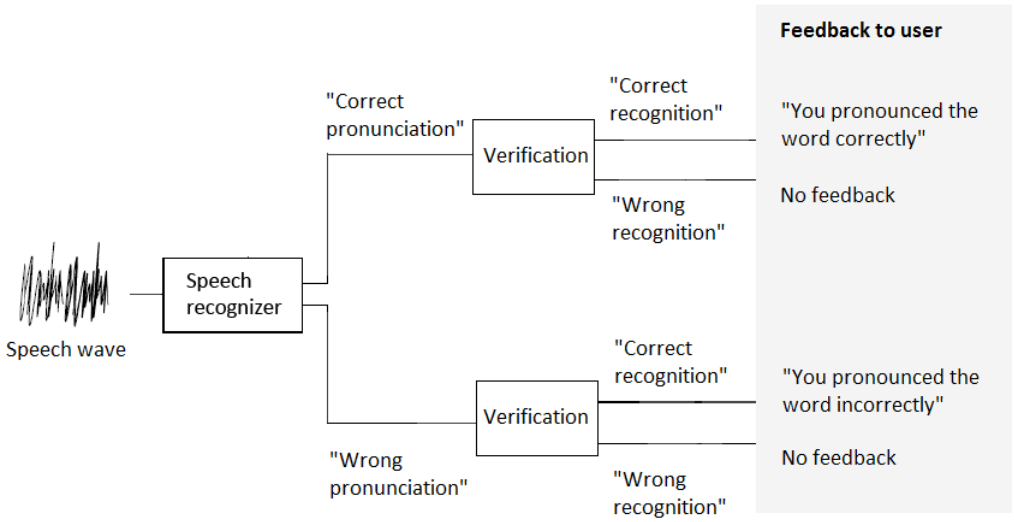

Figure 2: Feedback based on verification results

The speech recognition and verification system consists of three main components: An acoustic preprocessing component, a speech recognizer and a verifier, as illustrated in fig. 3.

The acoustic preprocessing changes the representation of the speech wave from a waveform to a set of vectors containing either Mel-frequency cepstral coefficients (MFCCs) or linear prediction coefficients (LPCs). The speech recognizer performs HMM based speech recognition on the MFCCs, in other words, tries to find the correct sequence of phonemes. Finally, the recognized phoneme sequence is given to the verifier which extracts features from both the MFCCs and LPCs, and uses these

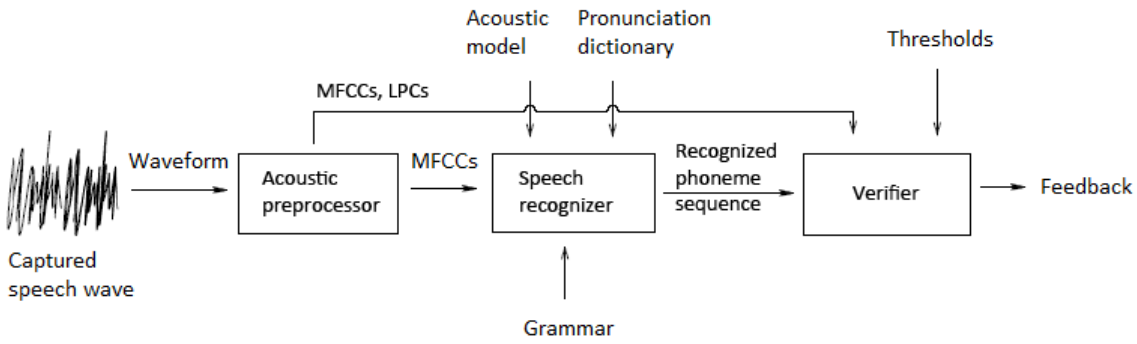

Figure 3: Overview of the speech recognition and verification system

features along with a predefined threshold to verify the correctness of the sequence of phonemes.

More detail on each component will be given in the following sections.

2.2 Phonetics

Just like written language is made up of different combinations of a finite set of signs, human speech is made up of different combinations of the set of sounds that the human vocal apparatus is able to produce. When these combinations are put into a system, and meaning is attached to them, they can be used to communicate information. This may imply that in order for two persons to communicate the same thing, they have to produce the same sounds. Since the vocal anatomy of each person is unique, just like fingerprints, it is not possible to do this. However, there may be enough commonalities between the two sound realizations that they can be perceived as the same. Phonetics is the study of these sounds, and how they can be produced, classified and transcribed [9].

The smallest unit in speech that holds meaning, i.e. that can distinguish one word from another, is a phoneme. In other words, phonemes are to spoken language as letters are to written language. Different phonetic alphabets, such as the International Phonetic Alphabet (IPA) [5] and Speech Assessment Methods Phonetic Alphabet (SAMPA) [4], have been created to standardize the representation of phonemes by assigning symbols to each of them.

Having symbols that represent each phoneme allows speech to be transcribed by writing down the symbols of the different phonemes in it. For example, a pronunciation of the word 'lyne' can be transcribed as /sil l y n @ sil/, using SAMPA and where 'sil' represents the silence before and after the word.

In Norwegian there are 9 vowels, and it is important to differentiate between long and short pronunciations of each vowel. Thus, 18 phonemes are needed to represent all vowel pronunciations. Each of these phonemes are listed in table 1, along with an example word.

The different sounds in speech can be separated into groups based on their manner of articulation. The list of different groups is quite long due to the many different techniques that can be used to produce sounds; however, this thesis will only inves-

Vowel	Length	Symbol	Word	Transcription
a	long	A:	hane	sil h A: n eh sil
a	short	A	Hanne	sil h A n eh sil
e	long	e:	lese	sil l e: s eh sil
e	short	e	lesse	sil l e s eh sil
i	long	i:	Lise	sil l i: s eh sil
i	short	i	lisse	sil l i s eh sil
o	long	u:	rode	sil r u: d eh sil
o	short	u	rodde	sil r u d eh sil
u	long	}:	mule	sil m }: l eh sil
u	short	}	mulle	sil m } l eh sil
y	long	y:	lyne	sil l y: n eh sil
y	short	y	lynne	sil l y n eh sil
æ	long	{:	være	sil v {: r eh sil
æ	short	{	værre	sil v { r eh sil
ø	long	2:	døme	sil d 2: m eh sil
ø	short	2	dømme	sil d 2 m eh sil
å	long	O:	måte	sil m O: t eh sil
å	short	O	måtte	sil m O t eh sil

Table 1: Phonetic alphabet (SAMPA) of Norwegian vowels

tigate one: The vowels.

Vowels are a subgroup of sonorant sounds, and are characterized by a free, non-turbulent, flow of air through an open vocal tract. Different vowels are produced by changing the position of the tongue or the shape of the lips, or both. This alters the shape of the vocal tract, which, in turn, changes the resonant frequencies.

The IPA has created a mapping between the different tongue placements and the vowels that are produced. This mapping is shown in figure 4, where the IPA symbols have been converted to SAMPA. The x-axis shows the position of the tongue, from front to back, and the y-axis shows how open the vocal tract is, i.e. how close the tongue is to the palate. The vowels may be rounded or unrounded depending on the shape of the lips. In the cases where two vowels are paired, the vowel to the right represents a rounded vowel, while the vowel to the left represents an unrounded one.

2.3 Speech Signal Representations

A representation of speech is a way to present the information in the speech in such a way that simmilarities and dissimilarities can be draw between different speech realizations. The different representations used in this thesis are: Waveform, spectral, spectrogram, linear predictive coding and Mel-Frequency spectrum.

The input to the system is a waveform representation of the sound wave, such as that in figure 5, produced by the student and recorded using a microphone. Each waveform varies greatly depending on what has been said, which person said it, in which environment it was said, and so forth. Due to these different types of

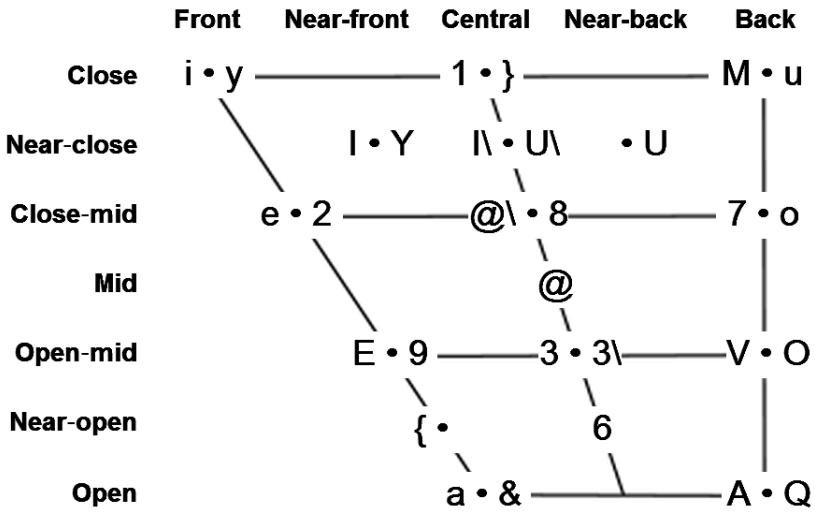

Figure 4: Mapping between vowels and the tongue positioning that creates them [3]. In the paired vowels, the vowel to the left represents an unrounded vowel, while the vowel to the right represents a rounded one.

variation, it is difficult for the ASR system to detect phonemes in the waveform directly. To do this properly, the system needs to find a different representation of the sound wave that is more suitable for speech recognition. This means that the representation should have small variations due to different speakers and different environments, and large variations due to different words or phonemes being spoken.

Spectrograms

Speech, or sound in general, is just a mixture of different frequency components. Because of this it is very useful, when evaluating speech, to look at the frequency content of it. This can be done by Fourier analysis, which finds the frequency spectrum of the signal. However, looking at the frequency content of the entire speech signal does not give much information. Instead, Fourier analysis should be done on smaller sections of the speech. This is known as short-time Fourier analysis, and will show how the frequency content changes over time, which is the general idea behind *spectrograms*.

A spectrogram is a 3D plot of the signal, where the three axes represent frequency, time and energy. However, they can also be 2D plots, like the one in figure 5, in which case time is on one axis and frequency on the other, while the energy at a given time and frequency is indicated by different colors on a color-scale. In figure 5, red represents high energy, and blue represents low energy. Looking at the spectrogram shows how the energy at different frequencies changes over time. The time periods with large energy variations indicate boundaries between the different phonemes in the speech. The longest phoneme, with the highest energy, is the vowel segment. Because each phoneme has its own distinctive look in a spectrogram, a phonetic expert can determine which phoneme was pronounced just by looking at it.

Calculating the spectrogram can be done in a few simple steps.

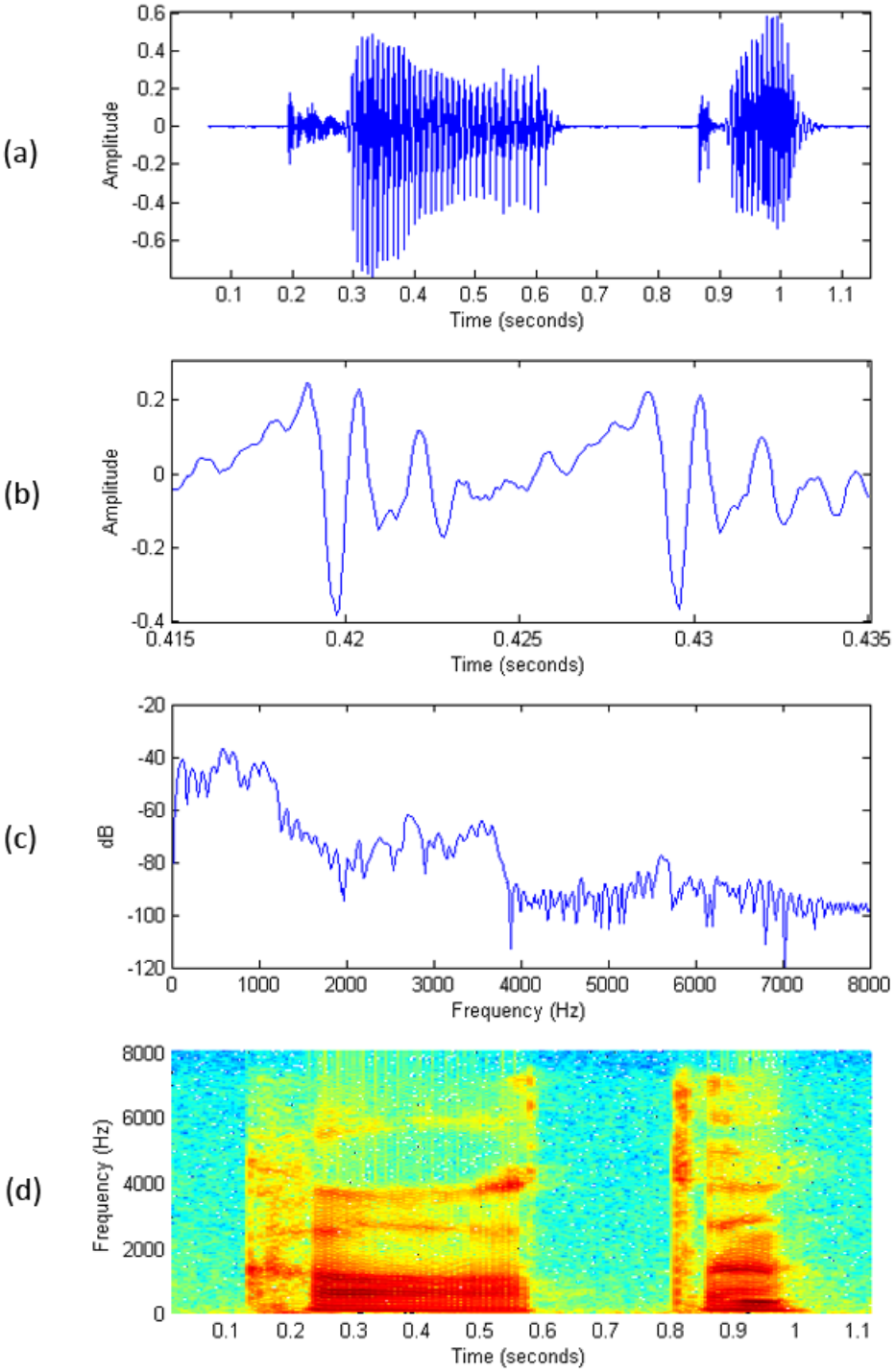

Figure 5: Different representations of speech. The speech was produced by a Norwegian pronouncing the word 'hane'. (a) Waveform. (b) Waveform of a 20 ms section. (c) Frequency spectrum of a 20 ms section. (d) Spectrogram

First, the entire speech signal is divided into several smaller sections, or frames. Selecting the right frame size is important. The Fourier analysis can only be calculated on each frame by assuming that the speech signal is stationary in that time, and thus the frame size has to be small enough for this assumption to be true. However, if the frame size is too small, the Fourier analysis will not give an accurate enough representation of the frequencies in the speech during that frame. In practice, frame sizes between 10 to 30 ms are used.

Next, the energy of the short-term Fourier transform is calculated. Figure 5 shows an example of a short-term signal, and its corresponding frequency spectrum. Finally, when the spectrum of all the frames have been calculated, they are put together to form the spectrogram.

Formants

Different vowels are created by changing the shape of the vocal tract and thereby changing the resonance frequencies. These resonant frequencies are whats know as formants, and are shown in the spectrograms as the areas with highest energy, i.e. the dark-red areas. The two first formants in any vowel phoneme, i.e. the two formants with lowest frequency, are denoted as F1 and F2, and just looking at these two formants can be enough to determine which vowel was pronounced. A list of some of the vowels and their corresponding F1 and F2 is shown in table 2.

Vowel	F1	F2
A	700	1150
e	500	2300
i	320	2500
u	320	800
y	320	1650
2	500	1500

Table 2: F1 and F2 for different vowels, captured from (CITE)

Formant calculation using LPC

Linear predictive coding (LPC) is a powerful method used in speech analysis. Chapter 6.3 in [9] explains the basic idea of this method, and how it can be used to estimate the formants in the speech. Once the coefficients of the LPC have been found, its spectrum can be plotted, as shown in figure 6. The peaks in the spectrum represents the formants in the speech. Selecting the right filter order is essential, and a rule of thumb in speech analysis is that the order should be equal to $2 + (Fs/1000)$, where Fs is the sampling frequency of the speech file.

Mel-Frequency Spectrum

The representation that is used in most ASR systems today is mel-frequency cepstrum (MFC). MFCCs are the coefficients that together make up an MFC. Unlike

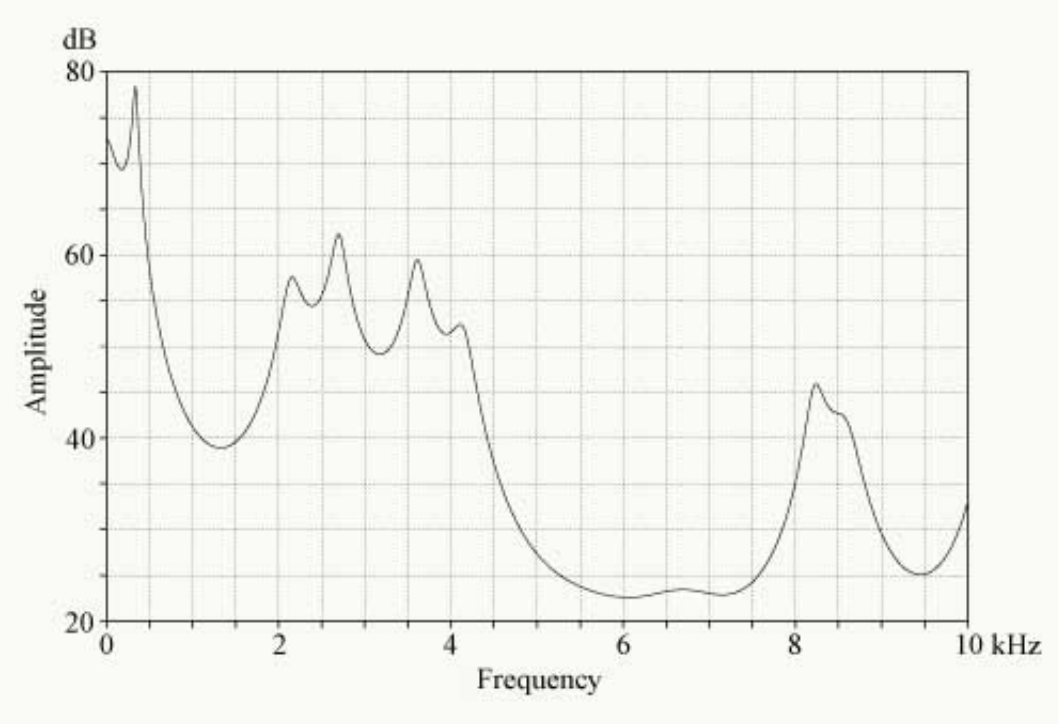

Figure 6: Example of a LPC spectrum for a given speach. Figure was captured from [2].

LPC, which models the way speech is produced, MFC models the way speech is perceived in the human ear. The MFCCs can be calculated in four steps, as shown in figure 7.

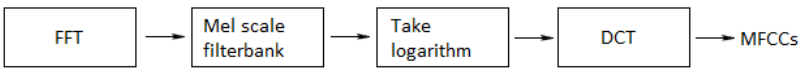

Figure 7: Four steps for calculating the MFCCs

Each step is explained below. Further details on MFC can be found in [11].

1. Take the Fourier transform of one frame.

2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.

3. Take the logs of the powers at each of the mel frequencies.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

2.4 Automatic Speech Recognition

This section will explain in more detail how the speech recognizer works. As can be seen from figure 3, the input to the recognizer is a vector containing MFCCs, a grammar, a pronunciation dictionary and an acoustic model. This section will explain what each of these inputs are, and how they are used to perform speech recognition.

Grammar

The grammar that goes into the speech recognizer is, simply put, a set of rules that determine in which order phonemes and words can be legally put together. For instance, if recognition were to be performed to determine the length of the vowel in an isolated word, a grammar such as that in figure 8 could be used. This grammar allows two legal combinations of phonemes: /sil l y n @ sil/ and /sil l y: n @ sil/.

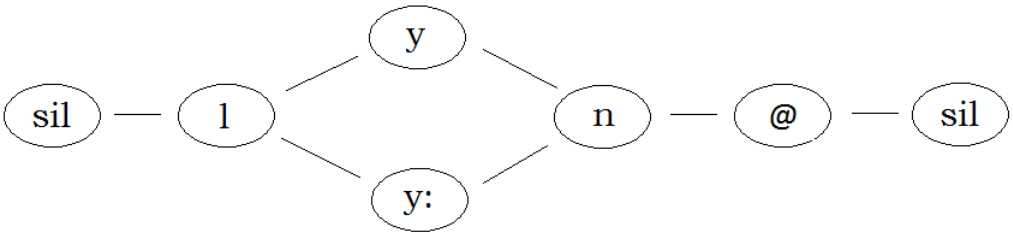

Figure 8: Example of a simple grammar

Forced alignment is technique where recognition is performed with a grammar that only has one legal combination of phonemes. This technique is usually applied to either find the segmentation boundaries between the phonemes, or to find the likelihood of the phoneme sequence, i.e. how well the sequence matches the speech.

Pronunciation Dictionary

The pronunciation dictionary is a mapping between words and the phonemes that make up the words. An example of a pronunciation dictionary for the two Norwegian words 'lyne' and 'lynne' is shown in table 3.

Word	Phoneme sequence
Lyne	/sil l y: n @ sil/
Lynne	/sil l y n @ sil/

Table 3: Pronunciation dictionary containing two words

Acoustic model: Hidden Markov Model

A detailed description of hidden Markov model (HMM) theory is presented by Rabiner [15]. This section will extract and present the theory from this article that is relevant to this report.

A Markov model is a finite-state system which may be described at any time as being in a set of N distinct states, $S_1, S_2, \cdots, S_N$. At regularly spaced discrete time instants, denoted $t = 1, 2, \cdots$, the system undergoes a change of state according to a set of probabilities associated with the state. The state at time t is denoted q_t. In general, finding the probability of the system being in a state q at time t, requires knowledge about all the predecessor states. For the case of a first order Markov model, however, this probability is only dependent on the preceding state, i.e.,

$$P(q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_k, \cdots) = P(q_t = S_i | q_{t-1} = S_j) \tag{1}$$

A HMM is a first-order Markov model where the state is only partially observable. In other words, the state does not correspond to a directly observable event, but the observation is rather a probabilistic function of the state. The observations correspond to the physical output of the system being modeled, and the sequence of observations is denoted $X = x_1, x_2, \cdots, x_M$.

The three parameters that make up a hidden Markov model, λ, are the transition probabilities, a_{ij}, the observation distributions, $b_i(x)$, and the initial state distributions, π_i. These are defined in eq. (2), (3), and (4).

$$a_{ij} = P(q_t = S_i | q_{t-1} = S_j) \tag{2}$$

$$b_i(x_m) = P(x_m | q_t = S_i) \tag{3}$$

$$\pi_i = P(q_1 = S_i) \tag{4}$$

There are three basic problems that needs to be solved in order for the HMM to be useful in real-world applications:

1. Given a observation sequence, X, and a model, λ, what is $P(X|\lambda)$, the probability of the observation sequence given the model?

2. Given a observation sequence, X, and a model, λ, what is the most likely state sequence Q^* that produces the observations?

3. How are the best estimates of the model parameters, λ, found?

The first problem can be solved by transforming the joint probability of X and Q, $P(X, Q|\lambda)$, using Bayes's rule, and summing over all possible state sequences Q. This is shown in eq. (5). The problem can be solved using the Forward algorithm, and solving this problem means that there is a way to see how well a given model matches a given observation sequence. This is measured by $P(X|\lambda)$, which is also known as the likelihood of the sequence.

$$P(X|\lambda) = \sum_{allQ} P(X, Q|\lambda) = \sum_{allQ} P(X|Q, \lambda) P(Q|\lambda) \tag{5}$$

The second problem is known as the decoding problem. Finding the most likely state sequence can be done using the maximum a posteriori (MAP) decision rule, as formulated in eq. (6). In other words, solving the decoding problem is a search problem. This is therefore often done with the Viterbi algorithm.

$$Q^* = \arg\max_Q(P(X, Q|\lambda)) = \arg\max_Q(P(X|Q, \lambda)P(Q|\lambda)) \tag{6}$$

Solving the final problem means that there is a way to automatically estimate the model parameters, given a set of training data. Information about how to apply the forward and Viterbi algorithm to solve problem 1 and 2, as well as information on how to solve problem 3, can be found in [15].

Use of HMM in ASR

HMMs are the most common acoustic models used in ASR. In these cases, the states in the HMMs represent the phonemes in a language, and the observations, x, are the MFCC vectors from the acoustic preprocessing.

Performing speech recognition, i.e. finding the correct phoneme sequence and the segmentation boundaries, is now the same as solving the decoding problem presented earlier in this section. This can be quite complicated if recognition is performed on continous speech, since, in these cases, the number of different state sequences can be infinate. For the isolated-word cases, however, the number of legal state sequences are more limited. The state sequences has to start and end with states representing silence, while the states in-between represents the phonemes that make up the word.

An example of a model for the Norwegian word 'lyne' is shown in fig. 9. The number of times a states changes back to itself represents the length of that phoneme, while the times where a state changes to a different state represents the segmentation boundaries between those two phonemes.

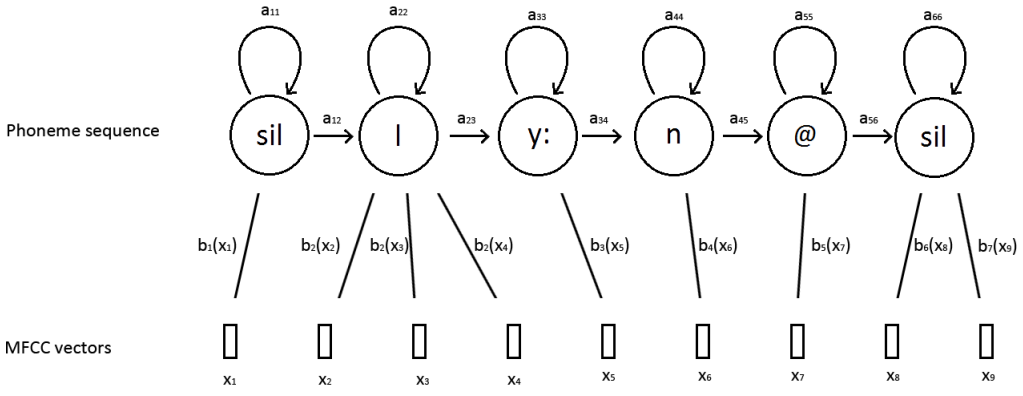

Figure 9: HMM model for the word 'lyne'

Since the observations are vectors, consisting of several MFCCs, the observation

distributions are not just 1-dimensional PDFs, but N-dimensional PDFs with N being the number of MFCCs in each vector.

2.5 Pronunciation Verification

The verification is a statistical hypothesis test, with the null hypothesis and the alternative hypothesis being:

H_0**:** The recognized phoneme sequence is correct.

H_1**:** The recognized phoneme sequence is incorrect.

In order to separate the two hypothesis, certain confidence measures are needed. These confidence measures are generated in such a way that a high confidence score suggests a high probability of the null hypothesis being true. A lower confidence score, on the other hand, suggests a high probability of the alternative hypothesis being true. This means that a threshold can be found which separates the two hypothesis. If the confidence score is higher than the threshold, the null hypothesis is accepted. If not, it is rejected.

Depending on which hypothesis is correct, and which hypothesis is selected, the verification may fall into one of four categories:

Correct Acceptance (CA): H_0 is correct, and H_0 is selected.

False Rejection (FR): H_0 is correct, but H_1 is selected.

Correct Rejection (CR): H_1 is correct, and H_1 is selected.

False Acceptance (FA): H_1 is correct, but H_0 is selected.

Knowing which of these categories each verification falls into is important to evaluate the performance of the verifier. The goal is to minimize the number of FRs and FAs, and maximize the number of CAs and CRs. This is achieved by finding good confidence measures that successfully separates the correct recognitions from the incorrect ones.

In addition, the numbers of FAs versus the numbers of FRs can be increased or decreased by adjusting the threshold. A high threshold produces more cases of FR, and less cases of FA. Consequently, a low threshold produces more cases of FA, and less cases of FR. It is therefore important to consider the environment in which the verifier will be used, when selecting the threshold. In some scenarios it can be beneficial to lower the number of FAs at the cost of a higher number of FRs, and vice versa. A common strategy, though, is to pick a threshold based on equal error numbers (EEN), where the amount of FAs is equal to the amount of FRs.

How these numbers are affected by an increase or decrease in the threshold, can be shown by a DET-curve, which is a plot of the number of FAs versus the number of FRs. An example of such a plot is shown in figure (CITE).

Features

There is a wide range of confidence measures that can be used to indicate the reliability of the result from a speech recognizer, most of which are presented and discussed in [10]. This section will look at the use of predictor features as a confidence measure.

As Jiang explains in [10], any feature can be called a predictor as long as its probabilistic distribution of correctly recognized words is distinct from that of misrecognized words. However, as he further explains, no predictor features are ideal, in the sense that they do not provide enough information to successfully divide correctly recognized words from the incorrectly recognized ones, in all cases. The overlap can be quite large even with the best predictor features. This is why many choose to combine several predictor features, in order to achieve a better performance. One way of combining predictor features is with the Fischer linear discriminant method (FLD). More information about this method can be found in [7].

The different features that were used in this thesis were log-likelihood ratio (LLR), vowel segment length (SL) and formant frequencies.

The vowel segment length can be calculated using the segmentation boundaries found during recognition, and is expressed as the number of frames in the vowel segment of the recognized phoneme sequence.

The log-likelihood (LL) of the vowel in a recognized sequence of phonemes, is also given by the speech recognizer. A forced alignment can be performed on an alternative phoneme sequence (e.g. the best competitor) to find the LL of the vowel in that sequence. The LLR is then given by the ratio between these two LL values, i.e. LLR equals the log-likelihood value of the vowel in the recognized phoneme sequence minus the log-likelihood value of the vowel in the alternative phoneme sequence. In other words, LLR = LL(recognized vowel) - LL(alternative vowel).

The final feature, the formants, can be calculated using LPC analysis, as explained in section 2.3.

3 Methods

This section will explain how the speech recognition and verification system was investigated. First, section 3.1 will discuss the research method that was used. Next, section 3.2 will present the speech database and the tools that were used. Finally, section 3.3 and 3.4 will explain the methods that were applied to perform verification on both vowel quality and quantity.

3.1 Research Method

The speach recognition and verification system evaluated in this thesis is not a system that has been tested alot before, especially not in Norwegian. Consequently, the research method used in this thesis had to be almost purely empirical.

First, the pronunciation quality in the speech files and the performance of the speech recognizer had to be evaluated. Since all of the speech was manually segmented and labeled beforehand, evaluation of pronunciation quality could simply be done by looking at the number of correct pronunciations versus the number of incorrect pronunciations, i.e. pronunciations where either the vowel quality or quantity was incorrect.

Performance of the speech recognizer was measured using WER, both in the case of vowel quality and quantity recognition. In the first case, erroneous recognition ment that the wrong vowel type was recognized. In the second case, erroneous recognition ment that the wrong vowel length was recognized. The WER was also calculated on speech from natives and non-natives separately, to see if the performance differentiated in these two groups. Since the recognizer had been trained on speech from natives, it was assumed that performance would be better on this group than the other.

Secondly, different approaches on how to implement this system, i.e. different verification strategies and features, had to be tested. This, also, had to be tested with regards to both vowel quantity and quality. Different strategies could be tested and evaluated by looking at how they differentiated in performance when the same features were used. Similarly, different features could be tested and evaluated by looking at how they differentiated in performance when the same strategy was used. Performance of the verifier was measured with the different numbers of CAs, FRs, CRs and FAs.

Finally, the successfulness of the system would depend on whether or not the verifier could improve the users learning experience when applied in a supposed CAPT environment. For every result that the verifier rejects, the user will have to repeat the same word once again. Thus, if too many results are rejected, the learning process may be slowed and the user may become frustrated. On the other hand, if too many of the incorrect results are accepted, the user will get confusing feedback, and this may also slow the learning process.

Challenges

One of the main challenges with this thesis was that all the tests that were done were very time consuming. As a result, not many tests could be completed within the

given time-frame. This meant that the different verification features and strategies that were used in each test had to be carefully selected, and therefore each feature and strategy was evaluated beforehand to see if they were worth testing.

3.2 Speech Database and Tools

The speech database was recorded in advance, and consisted of recorded speech from 13 speakers, 4 of them Norwegian, 3 Iranian, and 6 Chinese. Each of them were asked to repeat 18 words three times, giving 54 recordings per person, and 702 recordings in total. The words were both spelled out and pronounced by a teacher voice before the person had to repeat the word.

All the words that were used had the same consonants, but different vowels. They can all be transcribed as either k V: t @, or k V t @, where

$$\text{V} = \{\text{'A', 'e', 'i', 'u', '}\}\text{', 'y', '\{', '2', 'O'}\}$$

This means that only a some of these words are actual Norwegian words, while the others are nonsense words.

The reason for using these words is because the consonant before and after the vowel can impact the vowel itself. In other words, two vowels that are actually the same can look different if they are preceded or followed by different consonants. As a result, training a vowel on one set of preceding and following consonants, and then using a different set of consonants in testing can reduce the performance of the system. By having the same consonants in all the words, though, this problem is avoided.

Of the 702 recordings, 216 were from natives (Norwegians) and 486 from non-natives (Iranian and Chinese). The non-natives were all learning Norwegian, but considered to be in an early stage of the learning process. In addition, the non-natives all had first languages that was dissimilar to Norwegian.

Hidden Markov Model Toolkit

The Hidden Markov Model Toolkit (HTK) is a toolkit that is widely used in speech recognition research, and can be used for building HMM-based speech recognizers. A tutorial on the toolkit, including the basic principals of HHMs and how to use them in speech recognition, is presented in [1].

The tutorial shows that the toolkit consists of two major parts: One part for training, i.e. estimating the HMM parameters from training data, and one part for recognition, i.e. using the HMM parameters to find a transcription of unknown speech. The training of the HMM parameters was done prior to this thesis, on 20 hours of manuscript read speech from 900 speakers. Therefore, only the second part of the toolkit will be presented here, which includes the following tools: HParse, HCopy, HVite and HResults. A short presentation of these tools will follow below, however, more detailed information on each of them can be found in the tutorial.

HParse is a program which creates a word level lattice file from a grammar. The input syntaxFile is a text file containing the rules of the grammar on extended

Backus-Naur Form (EBNF), e.g. (sil k (A:|A) t sil). The output latFile is the lattice file which will later be used with HVite.

HCopy copies data files and saves them to a specified output file. While doing this, HCopy can also change the representation of the data. This was used to change the speech representation from a waveform to MFCCs, which means that it was the tool that was used in most of the acoustic preprocessing. The LPCs, however, were calculated in Matlab. Both the MFCCs and the LPCs were generated using a frame size of 25 ms, and a 10 ms step size.

HVite is the tool that was used for speech recognition, which also includes forced alignment. Simply put, HVite performs Viterbi recognition to match a speech file, represented by the MFCCs, against a network of HMMs as specified by the lattice file. It then outputs a transcription of the speech, along with the segmentation boundaries and likelihood values of each phoneme.

After recognition, *HResults*, was the tool that was used to evaluate the performance of the recognizer. This was done by comparing the transcription from the recognizer with the transcription from the manual annotation of the speech.

Training and Testing Environment

All the tools in HTK are invoked by entering command lines in a Unix shell, and all their inputs and outputs are HTK-formatted files. To do the many different operations that were needed in each training and testing session, Python scripts were developed so that each command would not have to be entered manually. File handling in Python, i.e. the different ways of writing and reading files, is both simple and fast, which makes this programming language well suited for this task.

Matlab is a programming language and computing environment that can be used to perform complex mathematical operations on large quantities of data. It was therefore used in thesis to do the FLD analyzis on the training data, to calculate the LPCs, and to create different plots of both the training data and the test data.

3.3 Verifying Vowel Quantity

There were two strategies that were used to evaluate the vowel quantity. The first one included information about the vowel type in the recognition, and assumed that the speaker had pronounced the correct vowel type. This meant that the recognizers job was to only determine whether the speech contained a long or short pronunciation of that vowel. Thus, if the speaker was supposed to say /k A: t @/, the grammar that was fed to the recognizer would be (sil k ($A:|A$) t @ sil).

The verifier should then either accept or reject the recognized phoneme sequence. If A: was recognized as the correct vowel, the null hypothesis and the alternative hypothesis would be

H_0**:** A: is correct

H_1**:** A: is incorrect.

The second strategy did not include information about vowel type in the recognition. This meant that the recognizer would only try to determine whether a long

or short vowel was pronounced, and ignore the vowel type completely. Therefore, if a long vowel was recognized as the correct vowel, the null and alternative hypothesis would be

H_0**:** V: is correct

H_1**:** V: is incorrect

where V represents the set of all vowel types.

With the exeption of different grammar used in the recognition, the training and testing procedure was the same for both of them. The two features that were used to test both strategies were LLR and SL.

Training

Training, in this scenario, is the process of finding a threshold that best separates the features in the correctly and incorrectly recognized speech. The training was done on each of the 18 words separately, which meant that 18 thresholds were found. In order to find the threshold for a given word, training data containing samples of both correct and incorrect pronunciations of that word is needed. The set of training data containing correct pronunciations is known as the positive set, or the positives, and the set of training data containing incorrect pronunciations is known as the negative set, or the negatives.

However, finding two such data sets was a problem, since only speech from the natives were used for training, and all of these speech files only contained correct pronunciations. The solution then became to use the pronunciations of the word with the same vowel type, but with a different vowel length, as examples of incorrect pronunciations of the word that was being trained. For instance, when training the word 'kate', native pronunciations of 'kate' were used as positives, and native pronunciations of 'katte' were used as negatives.

Since there were 4 native speakers and each of them repeated each of the 18 words 3 times, this meant that there would be 12 positives and 12 negatives, giving a total of 24 speech files used for training each word.

The procedure for extracting the features from each speech file, using the first verification strategy and training the word 'kate', was as follows:

- Perform forced alignment on the phoneme sequence /sil k A: t @ sil/
- Perform forced alignment on the phoneme sequence /sil k A t @ sil/
- Extract the segment boundaries of the vowel segment from the first and second forced alignment and calculate the segment lengths. These two values are known as SL1 and SL2, respectively.
- Perform two new forced alignments on each of the two vowel segments, and extract the 4 log-likelihood values that are created.
- Calculate LLR1 = LL(A:) - LL(A) for the first vowel segment.

- Calculate LLR2 = LL(A) - LL(A:) for the second vowel segment.

The procedure for the second verification strategy was identical, with the exeption that recognition was performed instead of forced alignement. The first recognition was performed with a grammar that included all the long vowels, to find the long vowel segment (SL1) and its log-likelihood value. The second recognition was performed with a grammar that included all the short vowels, which produced the short vowel segment (SL2) and its log-likelihood value.

After feature extraction was performed, FLD analysis on the features from the positives and negatives was performed, and a threshold was selected based on EEN.

Testing

Testing the verifier was done using speech produced by the Iranians and the Chinese. First, speech recognition was performed, which produced either a long or short vowel hypothesis. Next, the same features as those used in training was extracted from the speech. These features were then combined using data from the FLD analysis found during training, and compared to the threshold. The recognized vowel length was either accepted or rejected depending on whether the value was higher or lower than the threshold. The accepted or rejected result was then compared to what was actually said, indicated by the manual transcription of the speech. Based on this the current verification would fall into one of the following four categories: CA, FR, CR or FA

3.4 Verifying Vowel Quality

Verifying vowel quality is more complicated than evaluating vowel quantity. This is because with vowel quantity only two possible outcomes are possible during speech recognition. Either a short vowel is recognized or a long vowel is recognized. In vowel quality, however, there are nine outcomes, one for each of the vowel types. One way of verifying an outcome is to check if the confidence score generated by the recognized vowel is higher than the confidence score generated by the other vowels. Another way this can be done is by using some method to find the most likely alternative vowel, and compare this vowel to the recognized vowel. This was the strategy that was used for verifying vowel quality, and the null and alternative hypotheses could then be written as

H_0**:** Recognized vowel is correct

H_1**:** Best competitor vowel is correct

The verification features that were used were LLR and formant frequencies.

4 Results and Discussion

This section will present the results and discuss their implications.

4.1 Quantity Verification

4.1.1 Performance of the Speech Recognizer

The speech recognizer was the first part of the system that was evaluated. The performance of it could be determined by comparing transcriptions from the recognizer with the manually annoted transcription. This would indicate whether the recognition was correct or not, and the amount of correct recognitions versus the amount of incorrect recognitions would indicate either a good or a poor performance.

Since the two strategies used in quantity verification used different grammar in the recognition, this evaluation had to be done on both of them. The first strategy, where information on vowel type was excluded, is known as strategy 1. The other strategy, where information on vowel type was included, is known as strategy 2. Table 4, and table 5 show the numbers of correct and incorrect recognitions based on the nationality of the speakers, for strategy 1 and 2 respectively.

Speakers	Incorrect recognition	Correct recognition
Everyone	46 (6.55%)	656 (93.45%)
Norwegians	7 (3.24%)	209 (96.76%)
Iranians	5 (3.09%)	157 (96.91%)
Chinese	34 (10.49%)	290 (89.51%)

Table 4: Recognizer performance from strategy 1.

Speakers	Incorrect recognition	Correct recognition
Everyone	45 (6.41%)	657 (93.59%)
Norwegians	9 (4.17%)	207 (95.83%)
Iranians	8 (4.94%)	154 (93.83%)
Chinese	28 (8.64%)	296 (91.36%)

Table 5: Recognizer performance from strategy 2.

In addition to this, recognizer performance on differen vowel lengths was also evaluated. This is shown in table 6 and 7.

4.1.2 Evaluation of Training Data

Evaluation was also done on the training data that were used. Specifically, this meant looking at how well separated the features from the positives and negatives were. This can be illustrated by scatter plots, such as those in figure 10, 11, 12 and 13.

Speakers	Long vowel	Short vowel
Everyone	1.71%	11.40%
Norwegians	2.78%	3.70%
Iranians	0%	6.17%
Chinese	1.85%	19.14%

Table 6: Recognizer error rate on different vowel lengths, from strategy 1.

Speakers	Long vowel	Short vowel
Everyone	4.27%	8.55%
Norwegians	6.48%	1.85%
Iranians	6.17%	3.70%
Chinese	1.85%	15.43%

Table 7: Recognizer error rate on different vowel lengths, from strategy 2

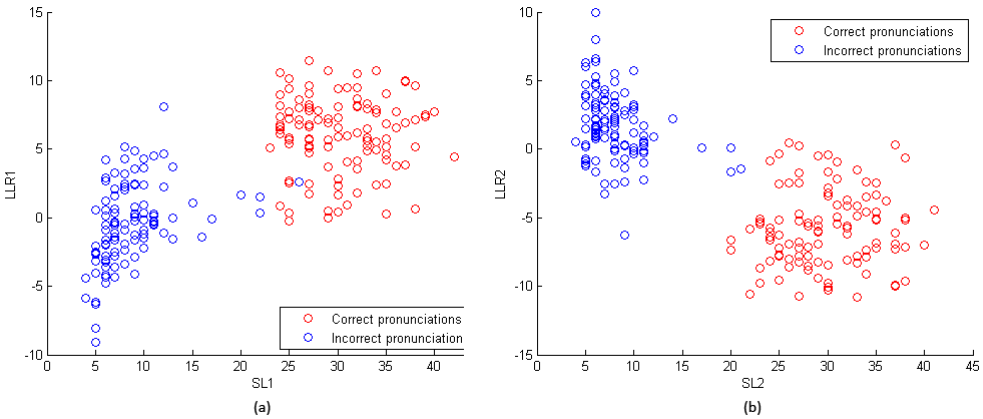

Figure 10: Scatter plots of features from the training data used to train all the long vowel words with strategy 1. Red circles show correct pronunciations, i.e. long vowel pronunciations, and blue circles show incorrect pronunciations, i.e. short vowel pronunciations. (a) Scatter plot of LLR1 and SL1. (b) Scatter plot of LLR2 and SL2.

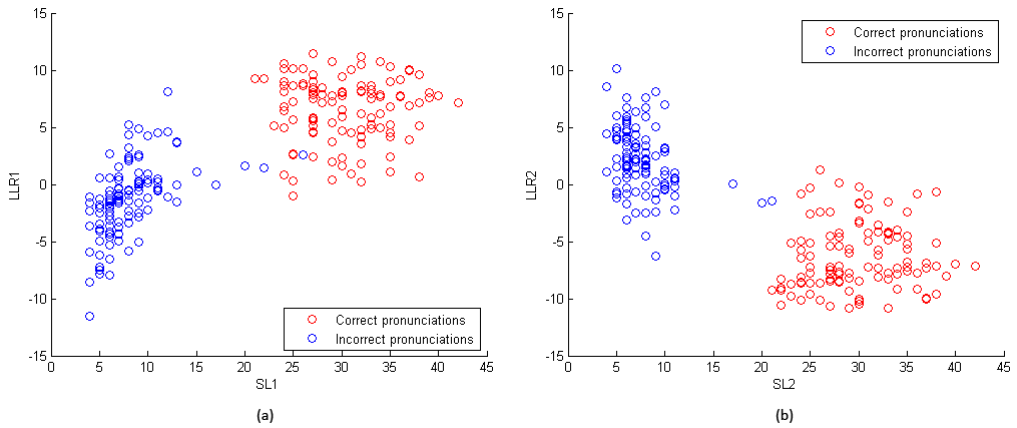

Figure 11: Scatter plots of features from the training data used to train all the long vowel words with strategy 2. Red circles show correct pronunciations, i.e. long vowel pronunciations, and blue circles show incorrect pronunciations, i.e. short vowel pronunciations. (a) Scatter plot of LLR1 and SL1. (b) Scatter plot of LLR2 and SL2.

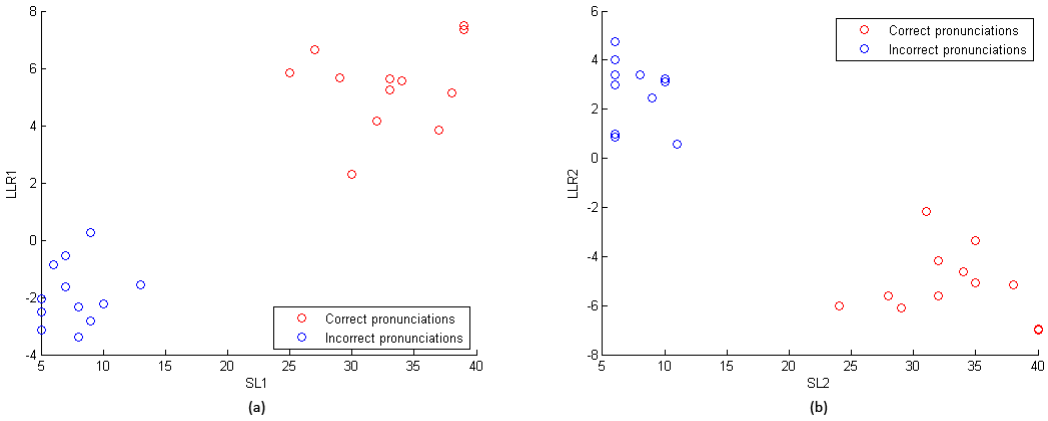

Figure 12: Scatter plots of features from the training data used to train the word 'kate' with strategy 1. Red circles show correct pronunciations of the word, and blue circles show incorrect pronunciations. (a) Scatter plot of LLR1 and SL1. (b) Scatter plot of LLR2 and SL2.

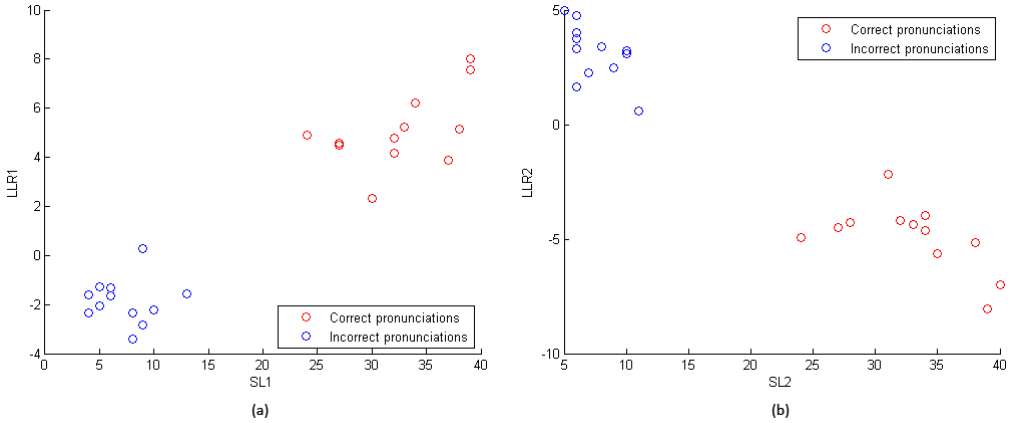

Figure 13: Scatter plots of features from the training data used to train the word 'kate' with strategy 2. Red circles show correct pronunciations of the word, and blue circles show incorrect pronunciations. (a) Scatter plot of LLR1 and SL1. (b) Scatter plot of LLR2 and SL2.

4.1.3 Test Results

The test results were evaluated by looking at the different numbers of CAs, FRs, CRs and FAs that were created due to different features and strategies being used. Table 8 shows these numbers for the different features used with strategy 1, and table 9 shows these numbers for the different features used with strategy 2.

Features	CA	FR	CR	FA
LLR	422	25	22	17
SL	438	9	26	13
LLR + SL	443	4	29	10

Table 8: Performance of verifier when strategy 1 was tested.

Features	CA	FR	CR	FA
LLR	407	43	16	20
SL	445	5	30	6
LLR + SL	444	5	30	6

Table 9: Performance of verifier when strategy 2 was tested.

DET-plots were also created to show how the threshold could be either increased or decreased to produce different numbers of FAs and FRs. Figure 14 shows DET-plots for strategy 1, and figure 15 shows DET-plots for strategy 2.

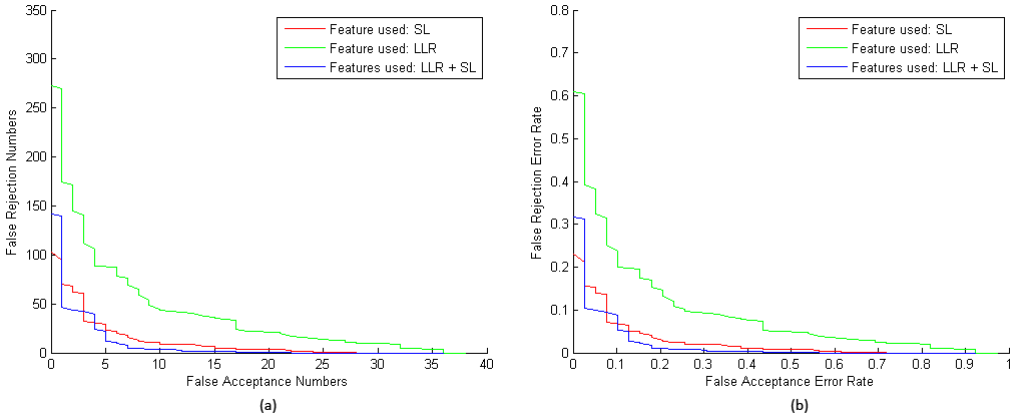

Figure 14: DET-plots for the different features used with strategy 1. (a) FA numbers vs. FR numbers. (b) FA error rate vs. FR error rate.

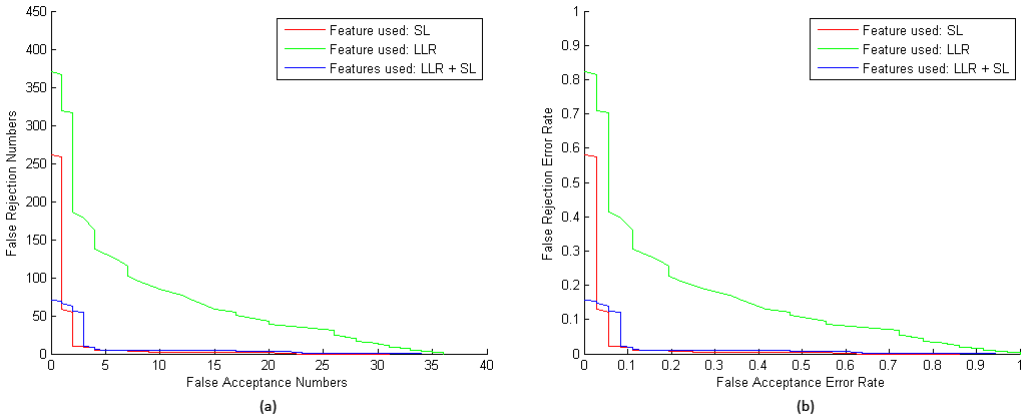

Figure 15: DET-plots for the different features used with strategy 2. (a) FA numbers vs. FR numbers. (b) FA error rate vs. FR error rate.

4.2 Quality Verification

4.2.1 Performance of the Speech Recognizer

Evaluation of recognizer performance was done in the same way as in section 4.1. This is shown in table 10.

Speakers	Incorrect recognition	Correct recognition
Everyone	250 (36.61%)	452 (64.39%)
Norwegians	45 (20.83%)	171 (79.17%)
Iranians	78 (48.15%)	84 (51.85%)
Chinese	127 (39.20%)	197 (60.80%)

Table 10: Recognizer performance

However, since there were a lot of pronunciation error related to vowel quality, evaluation was also done to see how this effected the recognizers performance. These results are shown in table 11.

Speakers	Correct pronunciation	Incorrect pronunciation
Everyone	32.19%	70.98%
Norwegians	20.83%	N/A
Iranians	45.14%	72.22%
Chinese	34.29%	70.46%

Table 11: Recognizer error rate on both correctly and incorrectly pronounced speech

4.2.2 Evaluation of Training Data

The features used in vowel quality verification were LLR1, LLR2, F1, F2 and F3. Figure 16 shows a scatter plot of F1 and F2 of the different vowel pronunciations in the training data. Also scatter plots of F1 and F2, and LLR1 and LLR2, from the training data used to train the words ‘kyte’ and ‘kete’, are shown in figure 17 and 18.

4.2.3 Test Results

The test results are evaluated the same way as in section 4.1. Table 12 shows the different CA, FR, CR and FA numbers produced by different features, and figure 19 shows a DET-plot of these features.

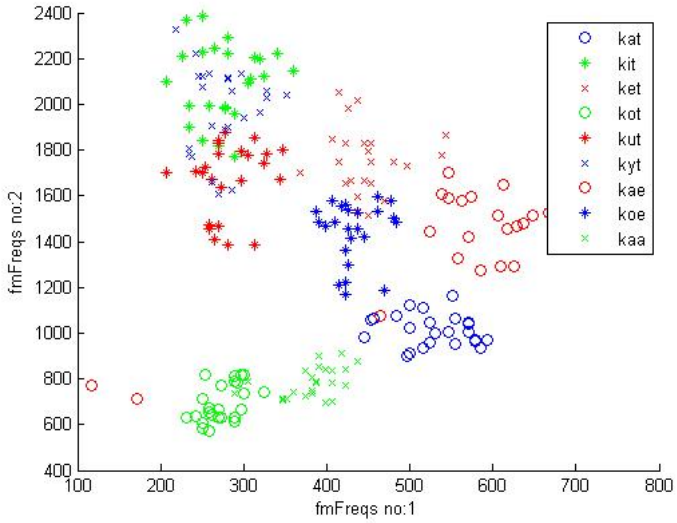

Figure 16: Scatter plot of F1 and F2 from the training data.

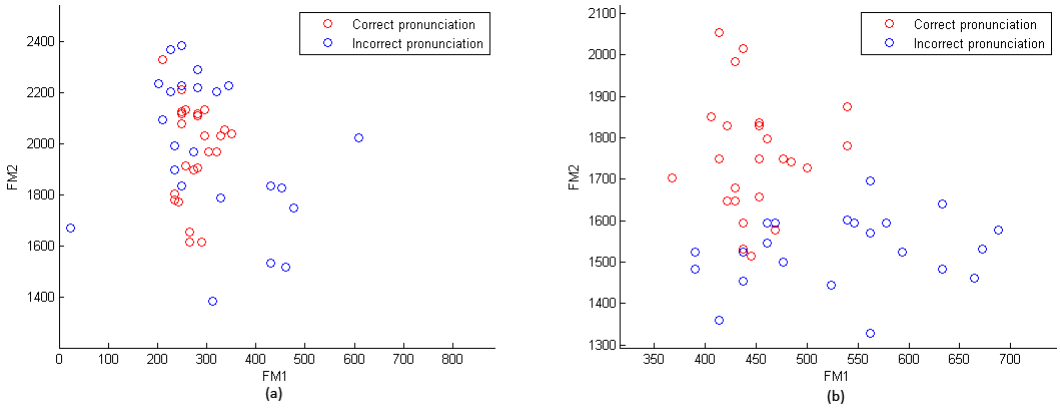

Figure 17: Scatter plots of F1 and F2 from the training data. Red circles show correct vowel pronunciations, and blue circles show incorrect pronunciations, i.e. pronunciations of another vowel. (a) Training data used to train the word 'kyte'. (b) Training data used to train the word 'kete'.

Features	CA	FR	CR	FA
LLR	249	51	54	132
3 formants	220	80	109	77
LLR + 2 formants	269	31	83	103
LLR + 3 formants	253	47	87	99

Table 12: Performance of verifier. Shown by CA, FR, CR and FA numbers for the different features used.

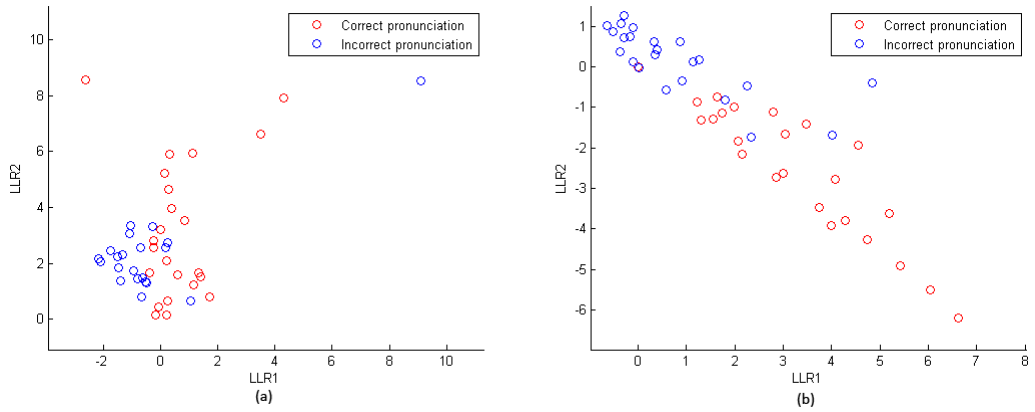

Figure 18: Scatter plots of LLR1 and LLR2 from the training data. Red circles show correct vowel pronunciations, and blue circles show incorrect pronunciations, i.e. pronunciations of another vowel. (a) Training data used to train the word 'kyte'.. (b) Training data used to train the word 'kete'.

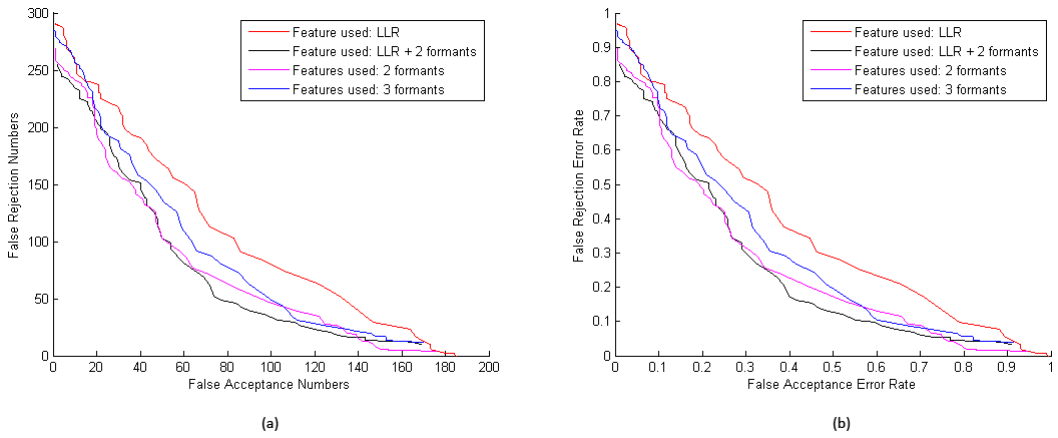

Figure 19: DET-plots for the different features used. (a) FA numbers vs. FR numbers. (b) FA error rate vs. FR error rate.

5 Conclusion

This thesis has discussed several ways of implementing the speech recognition and verification system presented in the introduction, and the goal was to evaluate whether this system could be used to evaluate the vowel quality and quantity pronunciations of second language learners that are trying to learn Norwegian. Three different verification strategies were presented, two of which were used to verify the quantity, and one that was used to verify the quality. The verification features that were used included the log-likelihood ratio, vowel segment length and formant frequencies.

The results showed that the recognizer performed well in determining the correct vowel quality, with error rates as low as 3.24 %, 3.09 % and 10.49 % for the Norwegian, Iranian and Chinese speakers, respectively. The combined error rate of the Iranian and Chinese speech, when strategy 2 was used, was 7.5 %. After verification, using both LLR and SL as verification features, this error rate was reduced to 1.35 %, by rejecting 7.2 % of the results. This indicates that the system can be usefull when evaluating vowel quantity.

Evaluating vowel quality, however, proved to be more difficult. In this case both the speech recognizer and the verifier performed poorly. The recognizers error was as high as 42.1 % when recognition was performed on speech from the Iranians and the Chinese. After verification, the error rate was reduced to 27,7 % by rejecting 23.5 % of the results. This is clearly not good enough, since the user of the system would, in this case, get either erroneous feedback or no feedback at all about 50 % of the time. It could be argued that the performance of the verifier could be improved using more sofisticated strategies or more discriminative features, but even with a perfect verifier the results would be rejected 42.1 % of the time. Clearly, if such a system is to be used in the evaluation of vowel quality pronunciation the performance of both the speech recognizer and the verifier has to be improved significantly.

References

[1] Hidden markov model toolkit book. http://htk.eng.cam.ac.uk/docs/docs.shtml.

[2] Introduction to phonetics. http://www.ling.upenn.edu/courses/Fall_2008/ling520/lab4/lab4.html.

[3] Multimedia ipa and sampa chart. http://www.lfsag.unito.it/ipa/index_en.html.

[4] Speech assessment methods phonetic alphabet. http://en.wikipedia.org/wiki/Speech_Assessment_Methods_Phonetic_Alphabet.

[5] International Phonetic Associtation. Ipa: International phonetic associtation. http://www.langsci.ucl.ac.uk/ipa/.

[6] Auralog. Tell me more. http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm.

[7] S. Balakrishnama and A. Ganapathiraju. Linear discriminant analysis - a brief tutorial. Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University.

[8] T. K. Hansen. Computer assisted pronunciation training: The four 'k's of feedback. In *Current Developments in Technology-Assisted Education*, Seville, Spain, November 2006.

[9] Xuedong Huiang, Alex Acero, and Hsiao-Wuen Hon. *Spoken language processing. A guide to theory, algorithm, and system development.* Prentice Hall PTR, 2002.

[10] Hui Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45, April 2005.

[11] Aldebaro Klautau. The mfcc. http://www.cic.unb.br/~lamar/te073/Aulas/mfcc.pdf.

[12] Ambra Neri, Catia Cucchiarini, and Wilhelmus Strik. Feedback in computer assisted pronunciation training: Technology push or demand pull? In *CALL professionals and the future of CALL research*, 2002.

[13] Ambra Neri, Catia Cucchiarini, and Wilhelmus Strik. Automatic speech recognition for second language learning: How and why it actually works. In *Proceedings of the 15th ICPhS Barcelona*, 2003.

[14] Ambra Neri, Catia Cucchiarini, and Wilhelmus Strik. Asr-based corrective feedback on pronunciation: does it really work? In *Interspeech, Pittsburg, USA*, 2006.

[15] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, February 1989.

[16] Todd W. Reesner. Tell me more-french, year = 2002, journal = CALICO, volume = 19, number = 2, pages = 419-428.