Institut für
Technische Informatik und
Kommunikationsnetze

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo

Automatic Pronunciation Checker

Kevin Jeisy

Master Thesis Spring 2015

Computer Engineering
and Networks Laboratory

Supervisors:

Dr. Beat Pfister
Tofigh Naghibi

August 22, 2015

Abstract

This Master thesis utilizes techniques from speech recognition to create an automatic pronunciation checker for language learning software. Second-language learners receive feedback based on their utterance. A long-term goal of this technology is to replace individual feedback from a human teacher with a language learning software.

The automatic pronunciation checker is realized by adapting the pattern matching algorithm that is usually applied in speech recognition. Since the classic implementation of pattern matching is speaker-dependent between speech signals in most cases, a neural network that was trained to be speaker-independent is used as a distance metric.

The parameters of the given approach are optimized using recordings of both correct and incorrect utterances. The results are evaluated to show the power but also the shortcomings of this implementation: while the simulations show that coarse errors are detected reliably, short deviations and shifted vowels often remain undetected with this approach.

Acknowledgements

I would like to thank my supervisors Dr. Beat Pfister and Tofigh Naghibi for their guidance and continuous support during this project. They made it possible to work on an interesting, wide topic.

Furthermore, I would like to thank all the volunteers that participated in the recording section of my thesis. Without their help, no results could have been produced.

Contents

Acronyms

AAE	Abstract Acoustic Element.
DTW	Dynamic Time Warping.
FN	False Negative.
FNR	False Negative Rate.
FP	False Positive.
FPR	False Positive Rate.
HMM	Hidden Markov Model.
MFCC	Mel Frequency Cepstral Coefficients.
MLP	Multilayer Perceptron.
NN	Neural Network.
PM	Pattern Matching.

1 Introduction

Speech recognition has been a focus of major technology companies in the past few years. They enabled users to dictate text and to execute commands using only their voice. Detection accuracy has improved substantially, making the technology useful for many types of purposes. Speech recognition is especially useful while in a car, since it does not require looking away from the road.

This thesis attempts to apply speech recognition in a different context: to assist in learning a second language. This technology should find an application in computer-based language courses, where learners would receive feedback on the quality of their pronunciation. Prior attempts were using a general-purpose speech detector: the recognized text was compared to the reference text as a rudimentary way of verifying pronunciation, achieving only low accuracy. The long-term goal in this area of research is to be able to give individual feedback on pronunciation without requiring a human teacher. In this case, this is done on the basis of pattern matching using dynamic time warping, while using a neural network as a distance metric.

1.1 Motivation

For second-language learners, it is easy to learn a second language in its written form without a human teacher. All it requires is dedication and a text book. It is not necessarily an advantage to receive input from a teacher, depending on learning style and preferences.

This is not the case for learning to speak a second language: It is not possible to learn correct pronunciation from a book. While a learner can use sound samples as a reference and repeat them, it is not given that the repetition of the learner is correct. Even if the utterance does sound correct to the person that is saying it, it is well possible that the person is unaware of certain aspects of the spoken language. For that, it is a requirement to get feedback on the utterance.

Usually, this means that a teacher has to be available to listen to the learner. In many cases this takes place in a classroom environment, leaving little time for individual feedback. The best way to improve the pronunciation of a learner is in one-on-one sessions with a private teacher. This thesis starts an attempt to reduce that requirement by providing each learner with a digital teacher that gives individual feedback.

1.2 Overview

In Chapter 2, the area of speech recognition is introduced. It is shown how the same techniques will be adapted and refined to be used as pronunciation checkers.

To be able to optimize the checkers and also to make statements about their quality, voice data has to be obtained. Such voice data was generated synthetically and collected from participants as shown in Chapter 3.

The data is applied to the checker in Chapter 4. The chapter shows the process of evaluating the output of the checker to optimize the parameters of the checker. For verification purposes, the optimization is performed on synthetic data first. After that, it is applied real voice data.

This report continues with an evaluation of the performance of the pronunciation checker (Chapter 5) and concludes this thesis with a summary and an outlook in Chapter 6.

2 Fundamentals

This Chapter covers the basics ideas and methods that are required for understanding the components of the pronunciation checker, as well as the way synthetic data is generated. In Section 2.1, the basic concepts of speech recognition are introduced. Why these approaches might not be suitable for pronunciation checking and how they could be adapted to better accommodate the specific requirements is discussed in Section 2.2. Basic metrics and algorithms that are relevant for understanding the experiments are mentioned in Section 2.3

2.1 Speech Recognition

In general, speech recognition is the process of turning spoken language into text. A division is made between two fundamentally different approaches of detecting spoken language (see [1, p.285]):

2.1.1 Pattern Matching

For each word of the vocabulary that needs to be detected, one or several reference samples are recorded. The test sample is compared to each of the reference samples and the best match is chosen as the detected word.

This approach primarily works for single words or (short) predefined sentences. The usual metric to calculate the similarity of two samples - the Euclidean distance between two Mel Frequency Cepstral Coefficients (MFCC, see Section 2.3.1) - is speaker dependent. This means that both the reference and the test sample need to be recorded by the same person in order to achieve accurate results. To resolve this issue, there have been efforts to create a speaker-independent distance metric using neural networks (NN) (see [2, p.17]). How a NN works and how it is applied to Pattern Matching (PM) is explained in Section 2.3.2.

The reference and the test signal usually do not have the exact same length. This can be due to the utterances having different speeds (either during the whole utterance or just in sections), or because one of the signals includes more silence at the beginning or ending. This is why PM usually requires the usage of Dynamic Time Warping (DTW): its purpose is to find a mapping between two sequences where each element of one sequence is assigned to one of the other. DTW is described in Section 2.3.3.

2.1.2 Statistical Approach

For each word in the dictionary, a statistical representation is calculated that includes both the distribution of MFCC vectors and the variability of duration. This can be done by using a collection of utterances of any given word. Since this usually requires many samples per word, it is not considered to be viable for a complete vocabulary. Instead, a statistical representation of

single phonemes or short sequences of them can be created; a word is then modeled as a chain of these elements. A test sample is checked against these statistical representations and the best fit is chosen. Usually, this is done by using the Forward algorithm on a hidden Markov model (HMM, see Section 2.3.4).

The big difference to the approach using PM is that instead of having a distinct reference sample that is used to calculate an absolute distance, a sequence of statistical representations of a word to calculate the probability of the test signal being the same word as the reference word is used. Using the statistical approach, it is possible to detect more than single words by expanding the HMM to accommodate several words to form sentences. Also, if the voices of multiple speakers are used to calculate these statistical representations, it is possible to create a speaker-independent approach. Additionally, PM is generally more sensitive to noise in the signal than the statistical approach.

2.2 Detecting Pronunciation Errors

The main scenario of this thesis is a learner who is uttering a single word (or a short sentence) given by the learning software. The software's task is to check this utterance and to mark it as either correct or incorrect. As an additional feature, it could denote the general position within the word where the learner did not pronounce correctly (if a mistake was made). For detecting these errors, it is required to adapt the common approaches from Section 2.1 as follows:

- In speech recognition, the uttered word is unknown. The best match based on the used approach has to be found. For detecting pronunciation errors, the uttered word is already known, which means that the metric that defined the quality of the match is not applicable here. Instead, a new way of describing the match within a given word has to be found. It should not be necessary to verify that the person did actually utter the correct word.

- While speech recognition would have to be optimized to work despite minor mispronunciations, the main focus here lies in the detection of those small errors.

- In the scenario of the learning software, it is important that a correctly uttered word will not be detected as a mispronunciation by the algorithm. If a learner receives a negative feedback for a correct utterance, the experience will be frustrating. Because of several factors (voice properties, dialect and accent, environment, the specific utterance) the algorithm will have to be forgiving of certain aspects. A good trade-off needs to be made to keep the detection rate of errors high (detecting an error where there is one) while keeping the false positive rate low (detecting an error where there is none).

2.2.1 Pattern Matching

For the scenario of a learning software, using an approach that uses PM is the most efficient way of creating a big set of learning units: since existing software usually includes the recording of

a word by a teacher anyway, a reference signal is already available. The issue of being speaker-dependent will have to be addressed in such a scenario without a doubt. Using a NN instead of the Euclidean distance could be a way to resolve this issue. In comparison to speech recognition, the local constraints for the DTW will have to be chosen more restrictively. The experiments of this project will build on this approach.

2.2.2 Statistical Approach

If the statistical approach for speech detection is adapted to be used for verifying pronunciation, the same statistical representations of phonemes can be used in an HMM. But instead of finding the word that has the maximum probability given a sequence of MFCC, the Viterbi algorithm is used to find the most probable path in the given word model. From this, a sequence of probabilities can be calculated, denoting how well each vector from the MFCC sequence is fitting into the word model. At positions with a big discrepancy, a mispronunciation is to be assumed.

Given that the statistical representations of the phonemes are accurate and have been created using the data of various speakers, it can be assumed that this approach is speaker-independent. It has to be noted that a given phoneme is not exactly the same in every language. This means that for each language, a new set of data would have to be gathered in order to create a complete representation of the elements in that language, making the approach language-dependent.

A solution has been proposed to resolve this: instead of dividing each language into its components, a set of so-called Abstract Acoustic Elements (AAE) can be extracted from a large set of spoken language, in different languages. AAE do not have a direct relation to phonemes; rather, they can be considered a representation of each possible component of spoken language. Any word could be constructed out of the set of AAE. By definition, using those elements would create a language-independent model of spoken language. They are not covered in this thesis, but they are recommended as a next step in the final chapter.

2.3 Further Basics

2.3.1 Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients (MFCC) are used in speech detection as a way to represent a signal. The data is analyzed in certain time intervals, where each time a MFCC vector is generated. The dimension of such a vector can be chosen depending on the application; also, the first and second derivative can be included. A sequence of MFCC vectors represents a spoken signal and can be used for direct comparison or statistical analysis (see [1, p.296]).

The properties of MFCC are similar to using the Discrete Fourier transform-Cepstrum, but research from psychoacoustics is incorporated into MFCC (see [1, p.90]). Its goal is to achieve resembling MFCC sequences for signals that are perceived to sound similar by humans. For example, instead of measuring the pitch of a tone by its frequency in hertz, the so-called Mel-scale is used.

2.3.2 Neural Networks

NN are an essential tool in Machine learning (see [3]). They are modeled after the functionality of a brain, meaning that they can be taught to perform different tasks. Given that enough resources are allocated to the NN, it could in theory simulate a human brain. In this thesis, a multilayer perceptron (MLP) is used to create a new distance metric for the comparison of two MFCC vectors, replacing the Euclidean distance.

The Euclidean distance is not speaker-independent. The main goal of using NN instead is to eliminate the component of the speaker from the distance metric. The details of how this was done can be seen in [4, p. 32].

2.3.3 Dynamic Time Warping

DTW provides the possibility of comparing two samples of spoken language even though they were not uttered at the same speed or with the same rhythm. Given two utterances in MFCC (s_1 and s_2), each vector of the sequence needs to be mapped to a vector of the other sequence. To achieve this, the distance of any MFCC vector from the one utterance to any MFCC vector of the other one (using a given distance function $dist$, example in Figure 1) is collected in the distance matrix d (see [1, p.308]).

$$d(i,j) = dist(s_1(i), s_2(j)) \tag{1}$$

4	1	1	1	1
6	3	1	1	3
3	0	2	2	0
1	2	4	4	2
	3	5	5	3

$$s_1 = \begin{pmatrix} 1 & 3 & 6 & 4 \end{pmatrix}, s_2 = \begin{pmatrix} 3 & 5 & 5 & 3 \end{pmatrix}, d(i,j) = |s_1(i) - s_2(j)|$$

Figure 1: distances d between the two given 1-dimensional sequences s_1 and s_2, using Euclidean distance.

To find the best mapping between s_1 and s_2, a procedure that finds the minimal accumulated distance has to be established, while $s_1(1)$ is mapped to $s_2(1)$ and $s_1(m)$ to $s_2(n)$, where m and n are the lengths of the sequences. This requires local restrictions that assure that the order of the MFCC vectors remains correct. A simple example for a set of local restrictions is given in Figure 2. Each restriction has a weight that denotes the multiplicand which the distance $d(i,j)$ will have to be multiplied with to calculate the accumulated distance. The higher the weight, the less attractive it is to use this specific restriction.

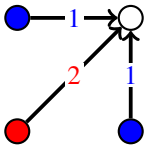


Figure 2: example of local restrictions: each colored dot denotes a starting point of a local restriction. The numbers of the same color correspond to the edge's weight.

DTW Algorithm By using the distance matrix and the local restrictions, the optimal path can be determined using the DTW algorithm (see [1, p.312]). In principle, for each point it is determined which of the local restrictions results in the path of the best accumulated distance (example in Figure 3). This optimal path is called the warping curve.

4	6	5	6	7
6	5	4	5	8
3	2	4	6	6
1	2	6	10	12
	3	5	5	3

Figure 3: accumulated distances D from Figure 1 using the local restrictions given by Figure 2. The warping curve is marked red, the total accumulated distance is 7 (top right).

The results of DTW depend strongly on the chosen local restrictions. In order to make a reasonable choice, the application of the DTW mapping has to be taken into account: while it is favorable to have larger tolerances for speech recognition in order to accommodate a wider variety of pronunciations of the same word, local restrictions for a pronunciation checker should be stricter since the correct pronunciation is desired.

2.3.4 Hidden Markov Models

HMM are an essential statistical model used in the context of machine learning (see [5]). In speech detection, HMM are used for a variety of applications. Basically, a statistical model (Markov process) is created to represent a word, sentences, or even a whole language. Using HMM, different questions can be answered:

- How probable is it that this utterance is the given word/sentence?
- What is the word/sentence that was spoken?
- How probable is it that the word was spoken in the given language?

In order to answer these questions, the Forward or Viterbi algorithms are applied to the given data.

3 Gathering Voice Data

To be able to optimize and evaluate the pronunciation checker, a large set of voice data is required. Gathering real voice data (Section 3.2) is always linked to a lot of effort; it is advisable to keep this to a minimum. Instead, a model to generate synthetic voice data is created (Section 3.1), which makes it easy to create lots of voice data with only minimal effort.

The approaches will use sequences of MFCC vectors as an input. This means that the artificially created voice data does not have to create sound data per se, but only those sequences. The real voice data will have to be converted to MFCC vectors. Chapter 4 will use this data for the conducted experiments.

It is important to find good examples of mispronunciations so that they represent the profiles of many learners. Section 3.3 introduces the notation of general pronunciation errors and gives an understanding of how few examples can be regarded as representative for big amounts of checks.

3.1 Synthetic Voice Data

For creating synthetic samples, HMM data of the statistical approach for speech recognition is used: since it requires a representation of each phoneme, it is possible to use it the other way around to create MFCC sequences. The given model provides 13-dimensional MFCC vectors. For each phoneme, three HMM states are modeled. Each state provides mean and variance in 13 dimensions as well as a state transition probability, which is used to denote the number of MFCC vectors each state should be generating. To create a word, these representations are connected in a Markov process (Figure 4 shows an example for /kar/).

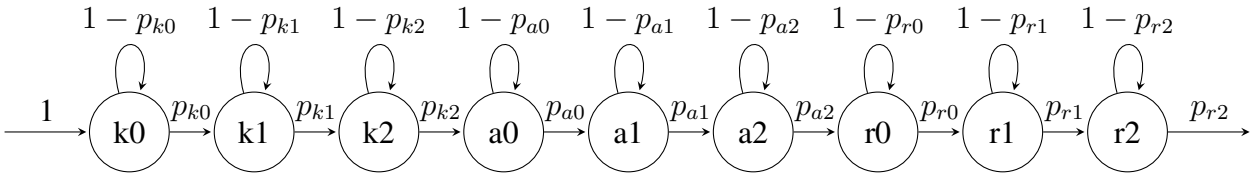


Figure 4: HMM model for the word car. Each state has a certain probability to be emitted again. Else, the model transitions to the next state.

Several problems arise when creating data using this Markov process:

- The duration of a phoneme becomes extremely arbitrary. Since only a state transition probability is given, a state can theoretically be emitted anything from one to infinite times. This creates an extremely high variability for the length and rhythm of words. While it is possible that a given word could have this high amount of variability, it is not something that should be considered in the context of pronunciation checking. For a learning software, the learner would be provided with a reference sample of the word that has to be pronounced, resulting in a similar utterance in terms of phoneme lengths.

To resolve this issue, both a minimum and a maximum number of emissions per state is introduced. Based on the state transition probability, it is possible to calculate the expected average number of emissions.

$$L_{expected} = \frac{1}{1 - Pr(X_{n+1})} \tag{2}$$

By using this as the reference length, the minimum and maximum numbers of emissions are set by multiplying with a configurable factor f.

$$L_{min} = \frac{L_{expected}}{f}, L_{max} = L_{expected} \cdot f \tag{3}$$

- All emissions are created independently of each other. If the statistical representations would have zero variance, this would not be a problem. But since there is variance, each emission will be different from the last one and therefore sound differently, even coming from the same state. For language synthesis, this does not make sense since most transitions between phonemes are gradual. Therefore, only one emission per state is generated. They are put into the middle of the emissions of the specific state. All the empty positions in the sequence are interpolated.

- Even though the statistical data is based on real voice samples, it cannot be guaranteed that the resulting MFCC sequences have a connection to real voice data. As a way of verifying this connection, a tool to create a sound signal out of the synthetically generated MFCC sequences was used. It turned out that for the generation of artificial MFCC sequences, the variance in the model was too high for voice data synthesis. By reducing the variability (by a factor $f = 4$), it was possible to create understandable voice samples.

While the synthesis of voice data can be used to verify pronunciation checking at a large scale, it has to be noted that this approach will not allow for testing of speaker-independence. It is not possible to synthesize different voices with only one set of HMM data. Real voice data will be recorded to test for speaker independence.

3.2 Real Voice Data

After verifying the functionality of the pronunciation checker using synthetic voice data, it needs to be verified how well the approach is working for real voice data. For this, a program is developed that allows the recording of words by participants. A set of 25 words was chosen; each participant recorded these words in both correct and incorrect pronunciations using that program (shown in Figure 5). For each recording, a reference signal is played back. The participants are asked to repeat the recording as accurately as possible.

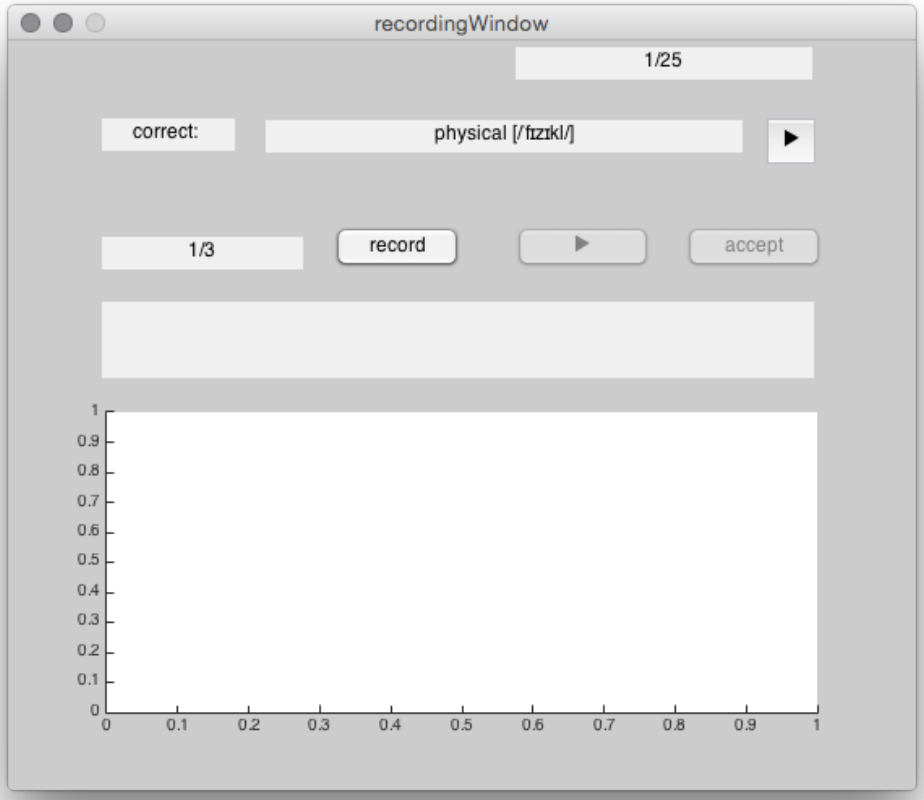


Figure 5: User Interface for recording real voice data.

Several mechanisms were put in place to make sure that the recordings can be used for the simulations:

- During the recording, it is checked if the signal level is too high or too low. Participants are asked to repeat the recording if there was an issue.
- The duration of the recording is compared against the reference signal. A deviation of 20% is allowed.
- After the recording session, each file is manually checked. There are a lot of problems that can disqualify the signal from being used, for example:
  - The wrong word being uttered
  - Improper signal boundaries, for example if a mouse click or a cough is recorded, or if part of the word is cut off.
  - Wrong pronunciation: for the experiment, it is important that the participant is uttering the words in the same way as the reference speaker. Since the pronunciations of the words were chosen so that different kinds of errors are contained within them, it is important to have consistent recordings.

Each recording was manually rated using the following categories:

- **Unusable**: This file does not represent the given word. It will be discarded completely for this experiment.
- **Poor**: While this is the correct utterance for the given word, it is not considered to be a good representation.
- **Good**: This is a good example of an utterance that a learner might do.
- **Reference**: A good recording that could be of a reference speaker.

After being rated, the recordings are converted to MFCC sequences that are going to be used by the checker. The conducted experiments can be found in Section 4.2.

3.3 Choosing Pronunciation Errors

As a reference for common pronunciation errors, a thesis from the University of Munich that collects common mistakes made by German-speaking persons when speaking English (see [6]) is considered. So-called phonologic rules are able to describe how the pronunciations are mutated.

$$A \longrightarrow B \quad / \quad D\_\_E$$

This term can be translated as 'A becomes B in the context of a preceding D and a subsequent E'. Even though there are more ways to mispronounce words, only the following general scenarios will be looked at:

1. Replacement:
   $$A \longrightarrow B \quad / \quad D\_\_E$$
   Vowel B is pronounced even though phoneme A would be correct.
2. Epenthesis:
   $$\emptyset \longrightarrow B \quad / \quad D\_\_E$$
   An additional phoneme is inserted into the pronunciation.
3. Deletion
   $$A \longrightarrow \emptyset \quad / \quad D\_\_E$$
   One phoneme is left out during the pronunciation of a word.

The phonologic rules will be used to create a large amount of synthetic samples: First, a dictionary is searched for words that contain certain patterns in their phonologic notation. Both correct and incorrect versions of the word are generated with a random component.

Additionally, there will be tests that check how well a change in the accentuation of a word can be detected. However, it is not possible to do this with synthetic data, since they do not provide enough flexibility to vary words in that way.

The simulation using real voice data was limited to the 25 recorded words. Each word and its mispronunciation was chosen so that it represents a different kind of error. The reason why this can be regarded as representative for a big amount of errors is because the pronunciation checker would behave similarly if that mistake was made in a different word. The DTW should align corresponding parts of the utterance, resulting in a match for all the parts except for those that contain the error - meaning that only the part with the error is relevant.

4 Experiments

In order to detect errors using PM, the learner's utterance of a given word is analyzed and converted to a sequence of MFCC vectors. It is then compared to a reference/teacher signal using DTW and a metric that denotes similarity between them (Euclidean distance for synthetic data, a NN for real voice data). For speech recognition, the accumulated distance would be used as a way of measuring similarity, usually dividing the distance by the length of the warping curve to standardize the output.

Since a wrong pronunciation usually only differs from the reference signal in certain parts of the utterance, the average difference (specifically the accumulated distance divided by the length of the warping path) will not be considerably bigger than in the case of a correct pronunciation. Instead, the distances $d(i, j)$ along the warping curve are considered. If the distance d is larger than a set threshold for longer than a given time, it is considered to be a mispronunciation. An example of this is shown in Figure 6.

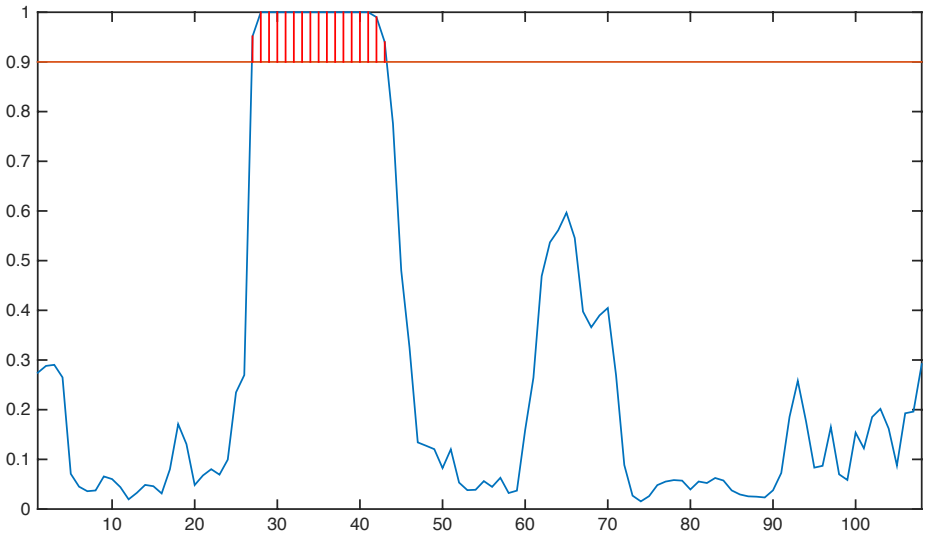


Figure 6: example plot of the local distance along the warping curve (blue). The orange line shows the threshold, the red lines show where the local distances are above that threshold. If the parameter of the minimum duration is 17 or lower, this sample would be considered a mispronunciation since there are 17 sequential samples with a distance above the threshold.

Section 4.1 shows how the performance is rated for synthetic voice data and how the parameters were optimized. Building on top of the results of the synthetic data, the same is done for real voice data (Section 4.2).

4.1 Synthetic Voice Data

As described in Section 3.1, MFCC sequences of words are created by using a statistical model of phonemes. For each kind of mispronunciation that will be tested for (Section 3.3), multiple correct and incorrect versions are generated. Two kinds of tests are conducted with this data:

- correct-correct analysis: onto each possible pair of correct MFCC sequences, DTW is applied. If a discrepancy is detected at any position, it is categorized as a false positive (FP).
- correct-incorrect analysis: each correct sequence is compared to each incorrect one. Only at (or near) the position of the mispronounced area, the local distances should be above the threshold for the given duration. If not, it is categorized as an FP as well. If no error is detected at the position of the mispronunciation, a false negative (FN) is noted.

Since synthetic voice data is not speaker-dependent, the Euclidean distance can be used as the distance function. Out of these calculations, a false positive rate (FPR) and a false negative rate (FNR) are calculated. These are used to evaluate the performance of different combinations of minimum threshold and duration, given certain local restrictions of DTW. To combine them to one single score, a weighting factor is introduced. Since it is considered to be worse to falsely correct a learner even if no mispronunciation was made, a low FPR will generally be weighted higher than a low FNR.

4.1.1 Optimizations

This section will describe the attempts that were made to optimize the error rate. Mainly, there are 4 parameters that are considered:

- Local Restrictions: This is the most versatile component of DTW. A wide range of possible combinations will be tested.
- Minimum peak height of local DTW distance
- Minimum peak duration of local DTW distance
- Weight of FPR vs. FNR: To limit the number of parameters, the weight of FPR was chosen to be 4 times as high as the one of FNR.

A given dictionary was searched for words that contain certain sequences of phonemes. From the chosen words, both correct and incorrect utterances were created to be used as a metric for the optimizations. The following shows the different mistakes that were used to evaluate the approaches:

- oː $\longrightarrow$ a͡u / ʔ__t
  (example: „automatic")
- u $\longrightarrow$ o͡y / ʔ__
  (example: „Euclid")
- sa͡i $\longrightarrow$ psy / __
  (example: „psycho")

- z $\longrightarrow$ x / __
  (example: „xylophone”)
- n $\longrightarrow$ pn / __
  (example: „pneumatic”)
- $\emptyset$ $\longrightarrow$ a / k__l
  (example: „practically”)

The specific words that contain these errors do not matter for the scenario of synthetic voice data, because the tested approach should be able to detect those errors in all contexts. As long as the analyzed feature set is big and diverse enough, a statement can be made about the quality of the approach. An issue that limits the general validity of the results is that the used phoneme model is language dependent. Second-language learners will often have the issue that their utterance of words is influenced by their first language. Amongst other things, they will use phonemes as if they spoke them in their first language. With the given data, it is not possible to create MFCC sequences that contain these kinds of errors. It is possible to analyze this using real voice data.

To find a good set of parameters, local restrictions were assumed and then iterated over the other aforementioned parameters. Based on the results, new local restrictions were developed in the hope of finding combinations that result in low FP and FN rates. The following paragraphs show the results of simulations with those local restrictions, starting with very basic ones up to a current optimum. The score s is calculated as

$$s = 100 \cdot (FPR + FNR \cdot w)$$

where w is the weight of the FNR in relation to the FPR (lower is better). It is set to 0.25 for this thesis.

4.1.2 Results

A reasonable configuration to start testing local constraints is to use a weight of 2 for proceeding normally (meaning if sample $s_1(i)$ is matched to $s_2(j)$, $s_1(i+1)$ will be matched to $s_2(j+1)$), and a weight of 1 to repeat one of the two samples (Figure 7a), yielding a score of 11.63. Optimizing those weights while keeping just those three possibilities resulted in a score of 9.43 (Figure 7b).

(a) (b)

Figure 7: basic local restrictions used to verify the functionality.

The restrictions based on Figure 7 may create an issue due to the fact that theoretically, one MFCC vector from signal s_1 can be mapped to any number of MFCC vectors of s_2. Such a warping curve would not match proper pronunciation. As an attempt to limit this, the local restrictions in Figure 8 do not allow the same sample to be used more than once. Their scores (17.88 and 11.19 when optimized, respectively) suggest that this does not lead to a better way of detecting errors. The reason for this is that it opens up the possibility to skip over samples that would not match well with the other sequence, leading to a worse detection rate. In an optimal scenario where the two sequences match perfectly, the warping curve would progress diagonally and the other constraints would never be applied. Deviating from the optimal scenario should be made as unattractive as possible while allowing slight discrepancies.

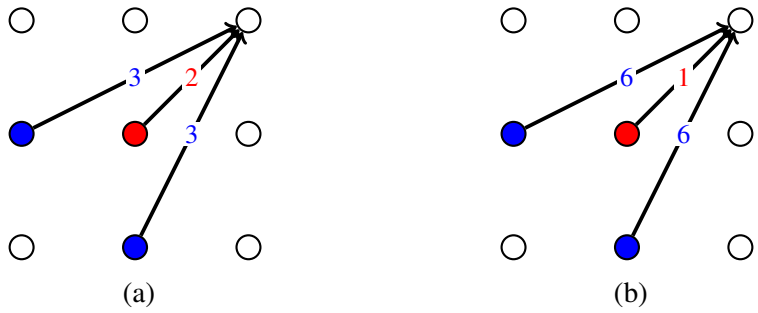


Figure 8: more confining local restrictions.

It is possible to combine these two kinds of approaches by chaining multiple restrictions so that proceeding horizontally or vertically only is not possible while still requiring every vector of the sequence to be taken into account. A direct conversion (Figure 9) does improve the results considerably (17.50 and 9.86) and delivers a baseline to expand on. The thought behind the upcoming approaches is that if the path is following the diagonal optimum for a longer time, deviating from it should be punished less.

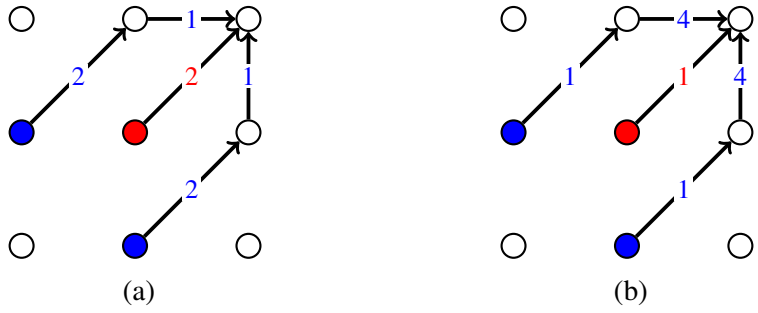


Figure 9: chained local restrictions.

Instead of having only one possibility for horizontal and vertical progression, an attempt is made to use several possibilities: the longer the warping curve is advancing diagonally, the less a horizontal or vertical segment is punished. Figure 10 shows such an approach; the score of 8.94 confirms that these local restrictions can improve the error rate. By iterating over every reasonable combination of weights, it is found that the local restrictions depicted in Figure 11 denote an optimum for the given scenario (score 8.36). The score consists of a FPR of 4.57%

and a FNR of 15%. An error is detected in this case if the local distance on the warping curve exceeds the threshold of 4.5 for the duration of 3 samples.

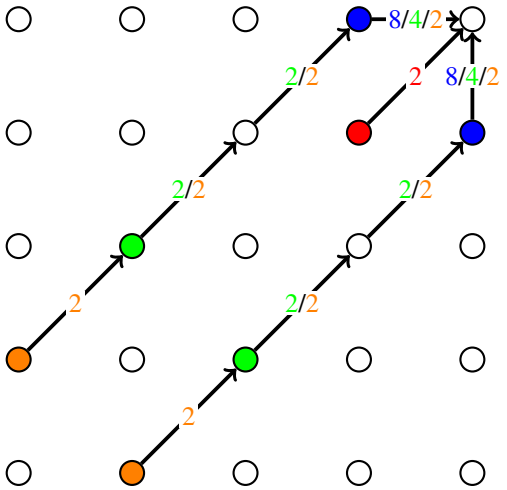


Figure 10: complex local restrictions.

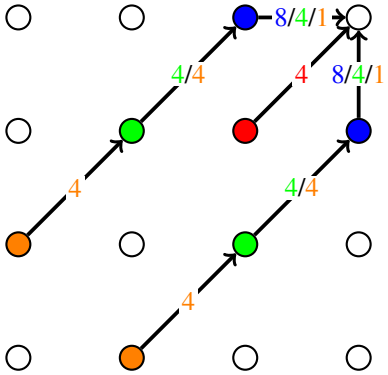


Figure 11: local restrictions optimized based on synthetic voice data.

When applied to a language trainer, this would mean that in 1 out of 20 cases, the checker would find a mispronunciation where there is none, and in 3 out of 20 cases it would miss an existing error. This can be regarded as a solid result. It has to be determined how well this can be matched using real voice data since they introduce different sources of error.

4.2 Real Voice Data

Even though the results of the experiments conducted with synthetic voice data provide a solid foundation for verifying the functioning of the approach and the idea of using PM for pronunciation error detection in general, they leave several factors untouched.

- The statistical data the synthetic voice data is based on cannot be used to test for speaker-independence. It would be possible using several sets of data, based on different speakers.

- Even if several sets of statistical data were available, it would be very hard to recreate the uniqueness of a person. A speaker might have a particular way of uttering certain constructs. This cannot be achieved with the given data set.
- There is only a limited set of mispronunciations that can be created using the statistical representations. For example, it is very hard to reproduce a shift in the accentuation of a word.

4.2.1 Voice Data

To be able to accommodate those factors, it is required to collect data from real persons. For that, a set of 25 words was put together (shown in Table 1). It covers various pronunciation errors that have different origins:

- **replacement**: A phoneme is replaced by another one. Detecting these errors can prove difficult, since the difference between two phonemes can be fluent. This means that an utterance can be ambiguous, especially when comparing recordings from two different persons.
- **epenthesis**: A phoneme is added to the correct pronunciation. These errors should be easier to detect since they usually change big parts of the word.
- **deletion**: Parts of a word are left out. This can happen if the speaker is skipping over a phoneme when speaking unclear. Usually, this does not change the rhythm of the word.
- **wrong accentuation**: Finally, cases were added where the sequence of phonemes are considered correct. Instead, the shift occurs in the wrong accentuation of the word.

Using the recording program introduced in Section 3.2, participants are asked to record each of those words multiple times, both in the correct and incorrect version. A reference signal is played back before each recording so that the participants know exactly how they should pronounce the word. In total, recordings of 9 male and 4 female participants were collected; for each word, 3 recordings of both correct and incorrect versions were made, resulting in 150 sound files per person. The recordings are manually rated based on the similarity to the given reference signal.

For the simulations, only recordings that are rated as *reference* (see Section 3.2) are going to be used as the reference signal. For the optimizations, only signals of ratings *good* and better will be considered.

4.2.2 Simulation Parameters

The simulation works similar to the one for synthetic voice data. From the recordings, MFCC sequences are extracted. Using a NN that was created as a way of having a speaker-independent

Table 1: List of mispronunciations used for recordings. The parts that are changed for the incorrect pronunciation are emphasized in bold face.

kind of error	word	correct pronunciation	incorrect pronunciation
replacement	ph**y**sical	fɪzɪkl	fyzɪkl
	h**ei**ght	ha͡ɪt	he͡ɪt
	sc**ie**nce	sa͡ɪəns	siːəns
	su**cc**ess	səkses	səkst͡ʃes
	automatic	ɔːtəmætɪk	a͡ʊtəmætɪk
	xylophone	za͡ɪləfəʊn	ksa͡ɪləfəʊn
	c**e**ment	sɪment	sement
	pron**u**nciation	prənʌnsɪeɪʃn	prəna͡ʊnsɪeɪʃn
epenthesis	ma**t**ure	mətʊər	mət͡ʃʊər
	com**f**ortable	kʌmftəbl	kʌmfərtəbl
	s**ui**t	suːt	suiːt
	practic**a**lly	præktɪklɪ	præktɪkəlɪ
	psychology	saɪkɑlədʒ͡ɪ	psaɪkɑlədʒ͡ɪ
	t**omb**	tuːm	tɑmb
	bus**i**ness	bɪznɪs	bɪzinɪs
	je**wel**ry	dʒ͡uːəlrɪ	dʒ͡uːwəlrɪ
deletion	lieut**e**nant	luːtenənt	luːtnənt
	prob**ab**ly	prɑːbəbli	prɑːbli
	en**tre**preneur	ɑntrəprən3r	ɑːnprən3r
	beau**ti**ful	bjuːtəfʊl	bjuːfʊl
	organi**za**tion	ɔːrgənaɪ͡zeɪʃn	ɔːrgənaɪʃn
wrong intonation	exec**u**tive	ɪgzekjʊtɪv	ɪgzekjʊːtɪv
	se**quence**	siːkwəns	səkwənz
	electronics	ɪlektrɑːnɪks	ɪːlektrɑnɪks
	technol**o**gy	teknɑːledʒ͡i	teknɑləːdʒ͡i

distance metric ([4]), PM is done on two MFCC sequences using the 88 different sets of local restrictions that are used to find an optimum. The relevant metric that is used for further experiments is the local distance along the warping curve. This sequence is analyzed; if the local distance is larger than a set threshold for longer than a given duration, the test MFCC sequence is considered incorrect (compare Figure 6).

Since the distance metric is a NN, the local distance is a value between 0 and 1. By iterating over this range in small steps (0.05) and iterating over reasonable minimal duration (1 to 15), a 15x20 matrix containing the information whether the pronunciation checker considers this as correct or incorrect is created.

For each word, every possible combination of reference sample to test sample is calculated and averaged. For each word, this results in one matrix of the FPR (when using correct test samples) and one of the FNR (when using incorrect test samples). Since the majority of the voice samples were recorded by male participants, the first part of the optimizations will be concerned with male voice samples only. As a second step, an optimization using the data of both male and female participants will be attempted. Discerning between male and female

samples is relevant because it is unknown how well the NN is able to eliminate the factor *gender* from the MFCC sequences.

By averaging the FPR and FNR matrices of all words, a metric is created that assesses the performance of a pronunciation checker whose task is to detect the errors of all 25 words. The same score as in the case of synthetic voice data is calculated, weighting the FPR four times as high as the FNR.

4.2.3 Optimizations

In this section, the potential and the limitations of automatic pronunciation checking using PM with a NN as a distance metric are explored. The optimizations will be conducted in an iterative behavior: an optimal parameter set is found by calculating the optimums of all possible parameters. This optimum is then analyzed and the limitations are evaluated. In the next iteration, a new optimum is calculated by ignoring the limitations that were found, hoping to achieve better results.

Male Participants As it was already mentioned, the scores are calculated by averaging all possible pairings of *reference* recordings and *good* and better test samples. Given the three parameters

- local restrictions of the DTW,
- error threshold of the local distance of the warping curve, and
- minimum duration of this error,

every possible combination is calculated and scored. Searching for the best score, the restrictions shown in Figure 12 using a threshold of 0.9 and a minimum duration of 9 are considered the best parameter set.

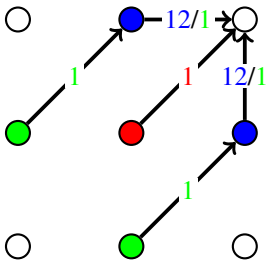


Figure 12: optimal local restrictions using all words for optimization.

The result is scored at 19.73 (FPR 3.74%, FNR 63.9%). Clearly, these numbers are a lot worse than the set baseline using synthetic data. Analyzing the results, it is revealed that the scores of the words

1. xylophone,
2. physical,
3. cement,
4. business,
5. probably, and
6. executive

are 25 or more. This score is worse than detecting every test as correct (FP 0%, FNR 100%, $0 + 0.25 \cdot 100 = 25$). Clearly, the detection does not provide usable results for those cases. Analyzing those words, it can be seen that the differences between the correct and incorrect pronunciation are only subtle:

- replacement (1-3): the difference between the phonemes of the correct and the incorrect pronunciation are only subtle. Except for word number 1, there are no clear borders between the correct and the replacing phoneme. For 1, it is possible that the hard k is partly cut off and therefore not considered.
- epenthesis (4) and deletion (5): the rhythm of the words did not change significantally, the inserted/deleted vowels only make up a very short part of the word. Since the local distance of the warping curve has to be above the threshold for longer than the given time, short variations are harder to detect. It is in the nature of DTW to allow for certain variations, which in this case leads to the wrong detection.
- wrong accentuation (6): even more significantly than the previous point, DTW simply eliminates the distance by putting the warping curve along the prolonged ʊ.

As an attempt to achieve better results, those words are not incorporated into the optimizations since there is a high probability that it will not be possible to detect the kinds of mispronunciations that they cover. Hoping that the optimizations will lead to a better detection for the remaining words, the simulations are run again without the mentioned words.

Ignoring these words improved the score to 14.95 (FPR 2.28%, FNR 50.7%). While the local restrictions are still the same (Figure 12), the minimum duration was reduced to 7. Checking for the worst performing words again, new bad-performing words are removed and the test is re-run. Removing the word *pronunciation* resulted in a slight improvement of the score (0.65). Looking at the scores of single words, none of them had a score of 25 or worse. But since the results of the simulations so far were not convincing, the remaining words with bad scores were checked for errors that are similar to the ones already removed.

- automatic (similar to xylophone)
- mature (similar to business)

- organization (similar to probably)
- technology (similar to executive)

Since those words all only did get scores of 22 or more, it was not realistic that the checker would be able to accommodate for those errors. Removing them resulted in a new optimum for the local restrictions (see Figure 13). The score decreased to 11.81 with an FPR of 1.73% and a FNR of 40.3%.

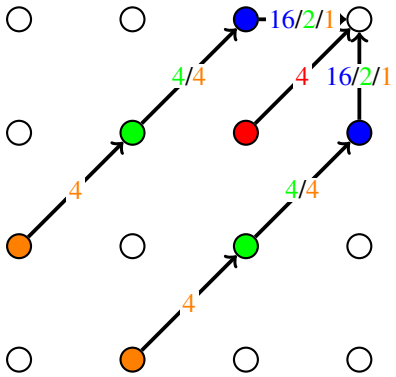


Figure 13: optimal local restrictions using a reduced word list.

While these new local restrictions improved the performance for the majority of words, they made detection of some errors worse (score 22 and up):

- height (similar to physical)
- jewelry, practically (similar to business)

By ignoring these errors, the optimal threshold and minimum duration parameters shifted (threshold from 0.8 to 0.9, minimum duration from 8 to 7), resulting in a score of 8.18 for the remaining words (FPR 1.19%, FNR 28.0%). Looking at the individual performances, all black-listed words have scores above 22, while none of the remaining words do. This configuration is going to be regarded as an optimum for the evaluation. Only 11 words remain:

- **replacement**: science, success
- **epenthesis**: suit, psychology, tomb, comfortable
- **deletion**: lieutenant, entrepreneur, beautiful
- **wrong accentuation**: sequence, electronics

Combined data For this optimization, a new way of calculating scores is introduced. Since there is an imbalance between the amount of male and female voice data, they are weighted differently. By calculating the aforementioned matrices of the FPR and FNR for

1. male-male,
2. male-female,
3. female-male and
4. female-female

datasets and weighting them the same (0.25), it is possible to calculate a case where the factor gender is ignored. Unfortunately, it turned out that for some words, no female recordings were classified as *reference*. This means that the combined FPR and FNR would not represent all words. Fortunately, when looking at the 11 words from the optimizations of male speakers only, just one word does not have the required data (*sequence*). For this analysis, the word choice is going to be limited to those ten words.

First, the optimal parameters for using only male speakers using the set of ten words is determined as a reference. As it turns out, the optimal local restrictions changed for this case. The new optimum can be seen in Figure 14. The score of 7.08 consists of an FPR of 3.11% and a FNR of 15.9%. An error is detected if the local distance is larger than 0.95 for the duration of 4.

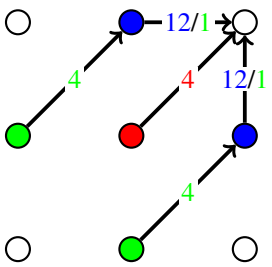


Figure 14: optimal local restrictions of the set of 10 words when using male training data.

Now that a baseline is established, an optimum is determined using the 4 mentioned combinations. Interestingly, a completely different set of local restrictions is found (Figure 15). The score of 13.61 (FPR 4.66%, FNR 35.8%) is considerably worse than the one using male voice samples only. Taking apart the four components of the score (1: 7.88, 2: 13.62, 3: 13.56, 4: 16.73), it can be seen that the performance of the detection is a lot better for male-male samples. If there was a problem with the checker when using reference and test speakers of different genders, only those scores would be worse. However, the female-female score is just as bad as the male-female case. Unfortunately, it is not possible to determine the origin of this with the limited amount of female voice samples (it could be because the detection for female voices is generally worse, or that the small sample size is not representative enough, or that this specific choice of words is better suited for male than for female voices).

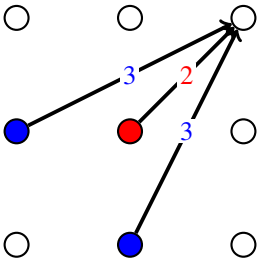


Figure 15: optimal local restrictions of the set of 10 words when using male and female training data.

Based on the issue described here, only male samples are going to be considered for the evaluation. But even though results are worse for the cases other than male-male, they are not worse by a magnitude, making them eligible for usage by the pronunciation checker.

5 Evaluation

This chapter will evaluate the performance of the pronunciation checker, using the optimizations shown in Chapter 4.2. Since the processing of the recordings showed that there were not enough usable recordings of female participants, the focus of the evaluation lies on recordings of male participants. A table of the determined parameters, and the score for each word is given in Appendix B.

There are several indicators that can be used to evaluate the pronunciation checker:

1. How well does it perform on the samples that were used to train it?
2. How well does it perform on new samples of the trained words?
3. What is the behavior for untrained words?
4. What kinds of errors can be detected, which ones does the checker have difficulties with?
5. Is there a pattern that influences the performance of the checker (long vs. short words, error in the middle vs. at the borders, etc.)

1. Training Samples Since the words for the training as well as the training samples themselves were chosen carefully, the results of only using the training samples turned out well. Only 1.2% of correctly pronounced words are not detected as such. This is well below the goal of 5%. In comparison, the FNR is fairly high, but as it was already mentioned, it is a less important metric for the application as a language trainer since this does not lead to a frustrating experience.

It needs to be said that this result carries only limited value. It is more interesting to see how well the checker performs on samples that were not used to train it.

2. New Samples of Training Words By using recordings that were classified as *poor*, samples that are similar to a person who is learning a new language can be checked. While the FPR tripled to 3.78% when using the *poor* samples, it still is well below the goal of 5%. The FNR did not change considerably. The score of 11.19 can be considered a success, since this means that a learner will be able have a good learning experience.

These scores still do not take into account that the words these samples are made of have been used to train the pronunciation checker. If a big and diverse enough sample size is used to determine the parameters of the checker, the performance should be the same for both trained and untrained words.

3. Samples of Untrained Words 25 words were recorded in the scope of this thesis, each covering one kind of mispronunciation. Out of these words, mispronunciations that are hard to detect using this approach were excluded from the optimizations, leading to 11 words used for training. The remaining 14 words were showing bad detection rates, both for the case where they were included in the optimizations and the case where they were not. They are not a good representation of a random new mispronunciation.

They can be used as bad-case examples though. Looking at the data of single words in Appendix B, it can be seen that the scores of the untrained words are all well above 20. But except for two cases, the high scores origin from a high FNR, meaning that no error was found. This is a clear indicator that the pronunciation checker is working as intended: while it is not able to detect every kind of pronunciation error, it does not create a bad experience for the learner. FP are a rare occurence: averaged over all recorded words, the FPR is 5.33%, which just misses the goal of 5%. Even when using the *poor* samples, the FPR remains within a usable range.

4. Kinds of Errors The following section will attempt to categorize the mispronunciations as an assistance in evaluating which kinds of errors are easier to detect. For reference, their pronunciations can be found in Table 1.

- replacement of a vowel (physical, height, cement and automatic)

  This kind of error is detected poorly. An explanation for this is that vowels do not have exact boundaries, the difference between one vowel to another is not clear, making it hard to discern between them. [7] shows an English vowel chart. While vowels have a general position in the shown plane, there is a big overlap between them. Additionally, the length of these vowels is fairly short. With the created pronunciation checker, it is not possible to give feedback on this kind of error.

- deletion of a syllable, without changing the rhythm of the word (entrepreneur, beautiful, lieutenant, probably and organization)

  This kind of error has resulted in both good and bad error detection. Comparing the words that performed well (*entrepreneur*, *beautiful*) with the bad performers (*probably*, *organization*), it is noticeable that the detection seems to work better if a hard consonant is contained within the deleted syllable (/bə/ and /zeɪ̯/ vs. /trə/ and /tə/).

- wrong accentuation of a word (electronics, sequence, technology, executive)

  Detecting wrong accentuation has proven difficult. While the two bad-performing examples (*technology* and *executive*) definitely sound wrong, the pronunciation checker is not able to detect an error. The reason for this is that the difference with those two examples only lies within the duration of one phoneme. The local restrictions of the DTW allow a mapping so that this small difference disappears.

  For the better two examples (*electronics* and *sequence*), changing the accentuation had a bigger impact on the word; the DTW did not eliminate all the differences in this case.

- insertion of a few phonemes, without changing the rhythm of the word (comfortable, business, practically, jewelry)

  The insertion of phonemes was generally not detected well. If the length of the insertion is very short, the DTW will build the warping path so that it is ignored. The only example where the insertion is longer (*comfortable*) was picked up well by the pronunciation checker.

- related to diphtongs (science, suit, pronunciation)

  While the replacement of regular vowels yielded bad results, the same cannot be said about diphtongs. The reason for this difference lies within the fact that the given examples have an emphasis on the vowel, making them generally longer. This is advantageous for the pronunciation checker. The word *pronunciation* does not have a strong stress on the modified vowel, which explains why it does not perform as well as the other two examples.

- For the rest of the words, no clear categorization could be made. What can be said is that words where only one phoneme was changed performed generally worse with the checker than words that had bigger components changed.

5. Indicators for Performance In this paragraph, it is analyzed if patterns can be derived from analyzing the data in Appendix B and the categories in Paragraph 4.

- short or long words: while the length of a word does not have a direct influence on the performance, longer words tend to generate more FP than short words. This can easily be explained: the longer the warping curve, the bigger the probability that the local distance is wrongly over the threshold. While this influence can be seen in the data, it is not something that needs to be addressed.

- modification in the middle or on the border: it is possible that errors which occur at the beginning or at the end of the utterance are harder to detect than errors within the word. The reason for that would be that the recording software uses an algorithm that cuts off the recording based on loudness. Even with small variations, the cutoff point can shift considerably, leading to inconsistencies. As it turns out, there is no recognizable influence from the position of the error.

- short or long modification: in Paragraph 4, there were hints that the checker performs better for mistakes that affect a longer portion of the word. Looking through the data, this holds true for most of the examples. The specifications of this pronunciation checker support this condition. It is unsure how this issue can be resolved using the given methods, since the categorization relies on a minimal error duration.

- vowel or consonant: the results show that it does not matter if the mistake is based on a vowel or on a consonant.

As a generalization, it can be said that the main factor that determines a successful detection of a mispronunciation lies in the length of the modification. This is a big drawback, since there

are many instances where only a short modification influences the perceived correctness of the word considerably. For example, this means in many cases it cannot be verified if the correct vowel was used. Because of this limitation, an important component of receiving feedback on pronunciation is missing from this checker.

6 Conclusions and Outlook

Using synthetically created voice data was a great approach to verify the functionality of the pronunciation checker. Would the approach not have worked, the recording of real voice data might not have been sensible. Also, it allowed to make a first evaluation of its performance for various mispronunciations. But since the model did not allow to create signals of different speakers, it was not useful to run more simulations on that setup.

When handling real voice data, using the Euclidean distance as the distance metric was no longer feasible. It was a requirement to use a speaker-independent metric. The approach of using dynamic time warping in combination with a neural network has proven to create a speaker-independent model that is able to verify pronunciation. The simulations have shown how well various errors can be detected; they also revealed a substantial drawback in that its performance is worse for short divergences from the correct pronunciation.

The simulations have shown that if new words are tested, the learner will still be able to have a good experience, because it is uncommon that correct pronunciations are marked as incorrect. In general, the checker is configured conservatively so that the learner only receives bad feedback if a mistake was made for certain.

This thesis did focus on voice data by male participants. All optimizations were made based on male teacher and learner signals. For verification, female voice data was recorded as well. It was shown that while the checker's performance is worse when including female voice data, it still gives usable feedback.

While it is possible that the optimizations differ depending on the language, the used procedure is language-independent. By using samples from various languages for the training of the pronunciation checker, it is possible to create a truly language-independent pronunciation checker.

A learner using this software will not be able to learn to speak the second language accent-free from just this feedback. Rather, it can be assured that correct pronunciations are learned by making sure they are repeated correctly. A human teacher is able to give feedback that is a lot more nuanced, which at this point still is the best way of learning to speak a language.

6.1 Future Work

- The results of this project can be used to create a language training software. While it will benefit from a bigger set of test words and mistakes to optimize the parameters of the checker, the functionality will stay the same. An open question is how helpful the feedback on coarse mistakes is in comparison to the fine mistakes this checker cannot accommodate.
- The approach of using Acoustic Abstract Elements was mentioned in Chapter 2. It provides a way to check pronunciation in a statistical approach. The collected data set can be used to compare the performance of using AAE in comparison to DTW.

References

[1] B. Pfister and T. Kaufmann, *Sprachverarbeitung - Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. 2008.

[2] S. Müller, *Sprachverarbeitungstechnologien für die computergestützte Sprechschulung*. 2008.

[3] C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[4] M. Gerber, *Speech Recognition Techniques for Languages with Limited Linguistic Resources*. ETH Zurich, 2011.

[5] N. Warakagoda, *Hidden Markov Models*. 2015.

[6] S. Biersack, *Systematische Aussprachefehler deutscher Muttersprachler im Englischen – Eine phonetisch-phonologische Bestandsaufnahme*. 2002.

[7] T. P. Szynalski, *English vowel chart*. 2015.

[8] B. Pfister, *Definition of the Phone Inventories to be Used for Mixed-Lingual TTS Synthesis*. 2010.

A English Phone inventory

IPA	ETHPA	Example	
ə	@	another	[əˈnʌðə]
əʊ	@_U	nose	[ˈnəʊz] [1]
æ	q	hat	[ˈhæt]
ɑ	A	got, frog	[ˈgɑt], [ˈfrɑg] [2]
ɑː	A:	stars	[ˈstɑːz] [1], [ˈstɑːrz] [2]
ʌ	V	cut, much	[ˈkʌt], [ˈmʌtʃ]
aɪ	a_I	rise	[ˈraɪz]
aʊ	a_U	about	[əˈbaʊt]
b	b	bin	[ˈbɪn]
ð	D	this, other	[ðɪs], [ˈʌðər]
d	d	din	[ˈdɪn]
dʒ	d_Z	Gin	[ˈdʒɪn]
ɜː	3:	bird, furs	[ˈbɜːd], [ˈfɜːz] [1]
ɜ	3	bird, furs	[ˈbɜrd], [ˈfɜrz] [2]
e	e	get	[ˈget]
eɪ	e_I	raise	[ˈreɪz]
ɛə	E_@	stairs	[ˈstɛəz] [1], [ˈstɛərz] [2]
f	f	fit	[ˈfɪt]
g	g	give, bag	[ˈgɪv], [ˈbæg]
h	h	hit	[ˈhɪt]
ɪ	I	witch	[ˈwɪtʃ]
iː	i:	ease	[ˈiːz]
ɪə	I_@	fears	[ˈfɪəz] [1], [ˈfɪərz] [2]
j	j	youth, yes	[ˈjuːθ], [ˈjes]
k	k	skat	[ˈskɑːt]
kʰ	k_h	kin	[ˈkʰɪn]
l	l	life, field	[ˈlaɪf], [ˈfiːld]
m	m	mean	[ˈmiːn]
ŋ	N	thing	[ˈθɪŋ]
n	n	fine, net	[ˈfaɪn], [ˈnet]
ɔː	O:	abroad	[əˈbrɔːd]
ɔɪ	O_I	noise	[ˈnɔɪz]
ɒ	Q	got, frog	[ˈgɒt], [ˈfrɒg] [1]
oʊ	o_U	nose	[ˈnoʊz] [2]
p	p	speed	[ˈspiːd]
pʰ	p_h	pin	[ˈpʰɪn]
r	r	ring, stress	[ˈrɪŋ], [ˈstres]
ʃ	S	shine, brush	[ˈʃaɪn], [ˈbrʌʃ]
s	s	sin, mouse	[ˈsɪn], [ˈmaʊs]
θ	T	thin, method	[ˈθɪn], [ˈmeθəd]

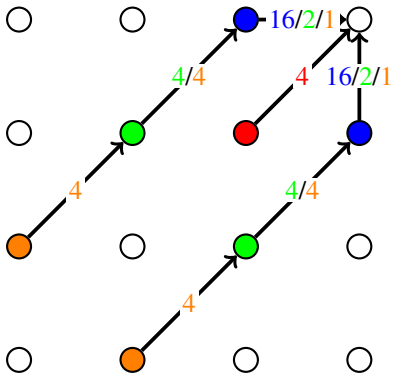
IPA	ETHPA	Example	
t	t	street	[ˈstriːt]
tʰ	t_h	time	[ˈtʰaɪm]
tʃ	t_S	chin	[ˈtʃɪn]
ʊ	U	book	[ˈbʊk]
uː	u:	lose	[ˈluːz]
ʊə	U_@	durable	[ˈdjʊərəbl]
v	v	very, heavy	[ˈverɪ], [ˈhevɪ]
w	w	well	[ˈwel]
x	x	loch	[ˈlɒx] [1]
ʒ	Z	vision	[ˈvɪʒən]
z	z	zoo, fees	[ˈzuː], [ˈfiːz]

[1] British English
[2] American English

Figure 16: English phone inventory in IPA and ETHPA notation. [8, p.5]

B Parameters and Scores of the Pronunciation Checker

- parameters:
  - local restrictions:



  - threshold: 0.9
  - minimum duration: 7

- scores:
  - trained words, samples: score 8.18, FPR 1.19%, FNR 28.0%
  - trained words, untrained samples (*poor*): score 11.19, FPR 3.78%, FNR 29.6%
  - all words (*good*): score 20.11, FPR 5.33%, FNR 59.1%
  - all words (*poor*): score 21.98, FPR 6.93%, FNR 60.2%

– score by word (*good*):

word	**score**	**FPR**	**FNR**
science	0	0.00%	0.00%
success	1.14	0.00%	4.55%
comfortable	1.56	0.00%	6.25%
entrepreneur	2.98	2.17%	3.23%
electronics	7.43	1.92%	22.0%
tomb	8.08	0.00%	32.3%
suit	9.48	0.00%	37.9%
beautiful	11.09	4.55%	26.2%
psychology	12.50	0.00%	50.0%
sequence	15.38	0.00%	61.5%
lieutenant	20.33	4.44%	63.5%
automatic	21.48	2.41%	76.3%
technology	22.50	0.00%	90.0%
mature	22.86	0.00%	91.4%
business	22.92	3.33%	78.3%
height	22.95	0.00%	91.8%
executive	23.15	0.00%	92.6%
probably	23.42	6.06%	69.4%
cement	25.00	0.00%	100%
practically	25.39	3.33%	88.2%
pronunciation	27.52	3.03%	98.0%
jewelry	27.86	4.00%	95.5%
organization	29.17	16.7%	50.0%
physical	30.97	6.38%	98.3%
xylophone	87.50	75.0%	50.0%

C Task Description

Institut für
Technische Informatik und
Kommunikationsnetze

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo

(MA-2015-28)

Task description for the master thesis

of

Mr. Kevin Jeisy

Supervisors: Dr. Beat Pfister
Tofigh Naghibi

Issue Date: February 23, 2015
Submission Date: August 22, 2015

Automatic Pronunciation Checker

Introduction

In computer-based language courses, pronunciation training is an important topic. For such a training, the computer has to decide if the language learner has uttered a certain word or phrase correctly. In the negative case the computer should indicate which phones have not been pronounced accurately enough.

In literature, there are many approaches to solve the problem of automatic pronunciation checking. Most methods are similar to speech recognition and thus are either based on some pattern matching (PM) or on statistical methods with hidden Markov models (HMM).

Pattern matching is advantageous in the context of language courses, because the method itself is language-independent and hence can be used for all languages of the generally large language portfolio of a course provider. The only requirement is the availability of references utterances of the words and phrases used in the pronunciation training. Such reference utterances are anyway available in computer-based courses, because the learner must have the possibility to hear how words are correctly pronounced.

The statistical approaches perform better because they include statistical models of the phonemes of the concerned language. However, the set of phonemes and their pronunciation varies from language to language. Therefore, a specific solution is needed for each language, at least a language-specific set of phoneme models.

Previous solutions to pronunciation error detection

In a previous master thesis, an approach based on PM has been investigated (see [1]). Standard PM, i.e. using MFCC features and Euclidean distance, is known to be speaker-dependent, which means that the matching process is accurate enough only if reference pattern and test pattern are from the same speaker.

In a language course, however, the reference patterns are from a teacher whereas the test patters are from a learner. In order to improve PM for this speaker-independent case, in [1] the Euclidean distance measure was replaced by a speaker-independently trained distance measure based on a neural network (NN). Actually, the NN was trained to estimate the posterior probability that two given feature vectors are from the same phoneme (see [2]). Although the results were much better for the NN-based approach than with the Euclidean distance measure, they were clearly not sufficient.

As can be seen from the literature, the statistical methods used for speech recognition have been successfully applied for pronunciation error detection as well (see e.g. [3] and [4]). Virtually without exception these systems are language-specific, however. Similar to speech recognizers, pronunciation error detectors are based on language-specific phonetic models.

Task of this thesis

In this master thesis, a new approach has to be investigated. This approach is based on statistical models of so-called abstract acoustic elements (AAE) which are not language-specific. The exactly same method is intended to be applied for every language.

AAEs are a kind of subword units similar to phonetic units, but have no linguistic meaning. They basically result from clustering of acoustic feature vectors that have been extracted from speech of many speakers and from various languages. In contrast to phonetic units, it is not easy to get a word model from AAE models. A solution to this problem has been proposed in [5]: An acoustic model for a word is derived from k utterances of that word by means of the k-dimensional Viterbi algorithm (see [6]). This algorithm determines the optimum sequence of AAE models for a small number of utterances of a word. Furthermore, the algorithm delivers the optimal alignment of these words and the word model.

Such an AAE-based word model can now be used to check the pronunciation of a given test word as follows: First, a forced Viterbi alignment is performed between the word model and the test word. With this alignment and those of the corresponding reference words (see paragraph above) we have the possibility to compare properties of individual segments of the test signal with the corresponding segments of the reference signals.

From this approach to pronunciation error detection, there are a number of questions arising such as:

- How many utterances of a word from how many speakers are necessary to get a good (e.g. speaker-independent) word model? Is it sufficient to use a unique model for a word or is better to use several word models, particularly when the utterances of the words divert considerably?

- What can be derived from the alignment information mentioned above? In particular, is it possible to detect wrong duration of phones from the duration of AAEs?
- How can the spectral quality be measured accurately? Do we need a NN-based distance measure like in [1] or are the segmental likelihoods from the word models (or the individual AAEs) usable instead?
- What kind of pronunciation errors should be detected in computer-aided pronunciation training? Are they language-specific? How can they be handled to keep the detector itself language-independent?
- Which speech features can be useful for detecting the relevant pronunciation errors?

These and further question need to be investigated in the framework of this master thesis.

Recommended procedure

This master project includes work in various topics. It is recommended to proceed as follows:

A. Getting acquainted with fundamentals (as far as necessary)

   Learn the fundamentals of HMMs. Read e.g. chapter 5 in [7] and do the corresponding laboratory exercises. Study the training of NNs in general (e.g. [8]) and of NN-based class verification in particular (see [2]).

B. Experiments with synthetic data

   An important class of pronunciation errors can be detected from spectral or timing properties of the speech signal. Such errors are e.g. insertion or deletion of a phoneme or strong deviation of duration or spectral shape. For a language-independent pronunciation error detector either PM or an AAE-based approach can be used. To estimate and compare the capability of these two approaches it may be suitable to perform the following experiments:

   - Design a generator for synthetic data which properties correspond to those of speech feature sequences, e.g. MFCCs. The generator should provide the possibility to synthesize feature sequences for words with and without pronunciation errors.
   - Realize a PM-based and an AAE-based detector for theses errors.
   - Test the detectors with synthetic data. Start with easy cases (e.g. with minor variability of the phonemes) and then successively increase the ambiguousness of the data.
   - Repeatedly refine the detectors and compare their performance.

C. Feedback for language learners

   Type and quantity of feedback to be given to a language learner may vary with his proficiency. The task of this thesis not to design this feedback, but to develop

the methods that allow to detect various kinds of pronunciation errors from learner utterances. In other words, it must be defined which types of errors have to be detected and which approaches and methods could be useful to detect them. Proceed as follows:

- Consult some literature (e.g. [9], [10] and [11]) to get a list of pronunciation errors that are important in the context of language learning. Assign each error type an importance score.
- Collect for each type of error a set of speech properties or features that could be used to automatically detect this error.
- Propose for each type of error possible error detectors and assign them a complexity score.
- Define a set of pronunciation error types and detection methods to be investigated in the framework of your thesis and discuss it with the supervisors. This will define the requirements of the pronunciation checker to be developed.

D. Development of pronunciation checker

This constitutes the main part of the master thesis. It includes the following works:

- For development and test of various algorithms, suitable speech data will be necessary. It is recommended to look for already available data and to reduce own collection of speech data to an absolute minimum. Before taking a decision, discuss your plan with your supervisors.
- Develop and test the necessary algorithms (based on the knowledge gained form part B and literature such as [10]). Note that a detector should provide for each detected error where it has been located in the speech signal, how severe the error is and how confident the detection result is. This additional information will be used later for designing the learner feedback which is not part of this thesis, however.

The work done and the attained results have to be documented in a report (see recommendations [12]) that has to be handed in as PDF document. Furthermore, two presentations have to be given: the first one will take place about two weeks after the start of the work and is meant to give a short overview of the task and the initial planning. The second one at the end of the project is expected to present the task, the work done and the achieved results in a sufficiently detailed way. The dates of the presentations will be announced later.

References

[1] S. Müller. Sprachverarbeitungstechnologien für die computergestützte Sprechschulung, 2008. Diplomarbeit am Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich (DA-2008-05).

[2] M. Gerber, T. Kaufmann, and B. Pfister. Perceptron-based class verification. In *Proceedings of NOLISP (ISCA Workshop on non linear speech processing)*, Paris, May 2007.

[3] C. Cucchiarini, H. Strik, and L. Boves. Using speech recognition technology to assess foreign speakers' pronunciation of Dutch. In *New Sounds 97: Proc. of the Third International Symposium on the Acquisition of Second-Language Speech*, pages 61–68, 1997.

[4] H. Strik, K. Truong, F. de Wet, and C. Cucchiarini. Comparing classifiers for pronunciation error detection. In *Proceedings of Interspeech*, pages 1837–1840, Antwerp (Belgium), 2007.

[5] M. Gerber. *Speech Recognition Techniques for Languages with Limited Linguistic Resources*. PhD thesis, No. 19507, Computer Engineering and Networks Laboratory, ETH Zurich, 2011.

[6] M. Gerber, T. Kaufmann, and B. Pfister. Extended Viterbi algorithm for optimized word HMMs. In *Proceedings of ICASSP*, pages 4932–4935, Prague (Czech Republic), May 2011.

[7] B. Pfister und T. Kaufmann. *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Verlag (ISBN: 978-3-540-75909-6), 2008.

[8] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[9] A. Neri, C. Cucchiarini, and H. Strik. Feedback in computer assisted pronunciation training: Technology push or demand pull? In *Proceedings of CALL Conference*, pages 179–188, 2002.

[10] K. Truong and A. Neri et al. Automatic detection of frequent pronunciation errors made by L2-learners. In *Interspeech*, pages 1345–1348, 2005.

[11] C. Cucchiarini, J. van Doremalen, and H. Strik. Practice and feedback in L2 speaking: An evaluation of the DISCO CALL system. In *Proceedings of Interspeech*, pages 779–782, 2012.

[12] B. Pfister. *Richtlinien für das Verfassen des Berichtes zu einer Semester- oder Master-Arbeit*. Institut TIK, ETH Zürich, Februar 2013.
(http://www.tik.ee.ethz.ch/spr/sada/richtlinien_bericht.pdf).

[13] B. Pfister. *Hinweise für die Präsentation der Semester- oder Master-Arbeit*. Institut TIK, ETH Zürich, Februar 2013.
(http://www.tik.ee.ethz.ch/spr/sada/hinweise_praesentation.pdf).

February 23, 2015