Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

# Detecting Strong Prosodic Events

Vanessa Hunziker

Janine Thoma

Semester Thesis SA-2015-06

Computer Engineering and Networks Laboratory
ETH Zurich

Supervisors:

Dr. Hui Liang
Dr. Beat Pfister

Professor:

Prof. Dr. L. Thiele

June 5, 2015

## Abstract

This work presents an approach for detecting strong prosodic events (i.e. emphasized words) in speech signals. The detection of emphasis is useful in speech-to-speech translation. It helps to ensure that post-translation prosody reflects speakers' original intentions, i.e. translated words corresponding to emphasized words in the input signal are also emphasized in the translated signal.

The steps taken in this work were split into three parts. Firstly, intensity, phone duration and fundamental frequency were extracted from input audio signals. Secondly, more compact features were derived from these three quantities. Finally, the derived features were utilized to train different classifiers, which were then evaluated. The necessary speech samples were taken form the multilingual SIWIS corpus.

This work's evaluation shows the best performing classifiers in combination with the chosen feature extraction are random forests for the languages German and English and support vector machines for French. Random forests achieved $F_1$ scores of 0.607 for German and 0.613 for English in out-of-bag classification. For French, an $F_1$ score of 0.606 was achieved by support vector machines in cross-validation tests.

# Contents


**1 Introduction** **3**

1.1 Motivation . . . . . 3

1.2 Approach . . . . . 4

**2 Implementation** **6**

2.1 Data Set . . . . . 6

2.2 Primary Features . . . . . 8

2.2.1 Duration . . . . . 8

2.2.2 Intensity . . . . . 9

2.2.3 Fundamental Frequency . . . . . 10

2.3 Derived Features . . . . . 11

2.3.1 Duration . . . . . 12

2.3.2 Intensity . . . . . 12

2.3.3 Fundamental Frequency . . . . . 13

2.3.4 Pause Duration . . . . . 16

2.4 Classification . . . . . 18

2.4.1 Classification Data Samples . . . . . 18

2.4.2 Support Vector Machines . . . . . 18

2.4.3 Random Forests . . . . . 19

2.4.4 Hidden Markov Models . . . . . 20

2.4.5 Neural Networks . . . . . 21

2.4.6 Parameter Optimization . . . . . 24

**3 Evaluation and Results** **25**

3.1 Feature Extraction . . . . . 25

3.1.1 Evaluation Methods . . . 25
3.1.2 Results . . . 25
3.1.3 Discussion . . . 35
3.2 Classification . . . 36
3.2.1 Evaluation Methods . . . 36
3.2.2 Results . . . 38
3.2.3 Discussion . . . 41

**4 Conclusion** **46**

**5 Future Work** **47**

**A Feature Histograms** **48**
A.1 Raw Features . . . 48
A.2 Derived Features . . . 50

**B Classifier Optimization** **52**
B.1 Support Vector Machines . . . 52
B.2 Random Forests . . . 53
B.3 Hidden Markov Models . . . 54
B.4 Neural Networks . . . 56

**C Phone Inventories** **58**

**D Task Description** **65**

# 1 Introduction

## 1.1 Motivation

Speech recognition transforms spoken language to written text. This process is bound to lose all of a speech signal's information that does not have a counterpart in the written language. One phenomenon exclusive to spoken language is prosody. This includes intonation, tone, stress and rhythm. Omitting prosodic information may lead to misunderstandings. Assuming that bold letters indicate a strong prosodic event, namely emphasis, the three sentences listed below have different meaning. Without emphasis the reason why Jill does not want to go to the movies with Jack tonight is unclear.

> Jill doesn't want **to go to the movies** with Jack tonight.
> (*Jill wants to do something else with Jack tonight.*)
>
> Jill doesn't want to go to the movies **with Jack** tonight.
> (*Jill would rather go with someone else or generally does not like Jack.*)
>
> Jill doesn't want to go to the movies with Jack **tonight**.
> (*Jill wants to go to the movies with Jack another day.*)

The above example shows prosody's dependence on pragmatic information. Other than that prosody is also influenced by semantics and syntax. In text-to-speech generation prosodic parameters are estimated using syntax analysis. For computers it is usually not possible to assess a text in its semantic and pragmatic aspects. This is why computer generated prosody does not measure up with human speech. In some cases, the ignorance of semantic or pragmatic information leads to wrongly generated prosody. For example, syntax-based prosody would not produce the emphases in the example sentences above. This problem is currently impossible to solve directly.

In speech-to-speech translation, however, there is a way to avoid logically incorrect prosody. The speech recognition part can be extended to additionally detect aspects of prosody in the original speech signal. Thus, the loss of prosodic information in speech recognition is reduced. After machine translation of the recognized text, the captured prosodic information can be combined with the syntax analysis-based prosody for speech generation.

The above described approach for better prosody in speech-to-speech translation relies on a suitable detector of prosodic information in the original signal. This work presents an approach for the detection of strong prosodic events (i.e. emphasized words). This task is non-trivial. While emphasis is easily audible to humans, it is hard to detect by speech signal analysis.

In the future it will be possible to incorporate this work's findings in speech-to-speech translation. The translated signal will thereafter be a better representation of speakers' original intentions. In the case of the example above, speech-to-speech translation will no longer introduce ambiguity. If the speaker originally emphasized *tonight*, this information will be carried on into the translated message.

## 1.2 Approach

To humans the difference between emphasized and neutrally spoken words is easily audible. However, the difference is less clearly visible in a recorded speech signal as shown in Figure 1.1. In this example the word *no* is emphasized in the second utterance. The first utterance does not contain any special emphases.

There is no direct way to detect the emphasized *no* directly from the audio recording as shown in Figure 1.1. Hence, any detection method must be of a more complicated nature. The approach taken in this work was split into three steps. Each of them is described briefly below.

**Raw feature extraction** According to [12] measurable prosodic quantities in a speech signal are fundamental frequency and intensity profile as well as phone and pause duration. The first two quantities can be calculated directly from a digital representation of a recorded speech signal. The last two require additional segmentation information. For this work, recorded audio signals and segmentation were provided. Section 2.2 describes the calculation of the four quantities in detail.

**Derived feature calculation** For classification purposes it is generally not feasible to directly use raw input features. It is common practice to extract more compact features first. Such derived features should contain as much of the input data's relevant information as possible while discarding redundancies. The process of derived feature calculation is described under Section 2.3. There is one derived feature for each of the four measurable prosodic quantities.

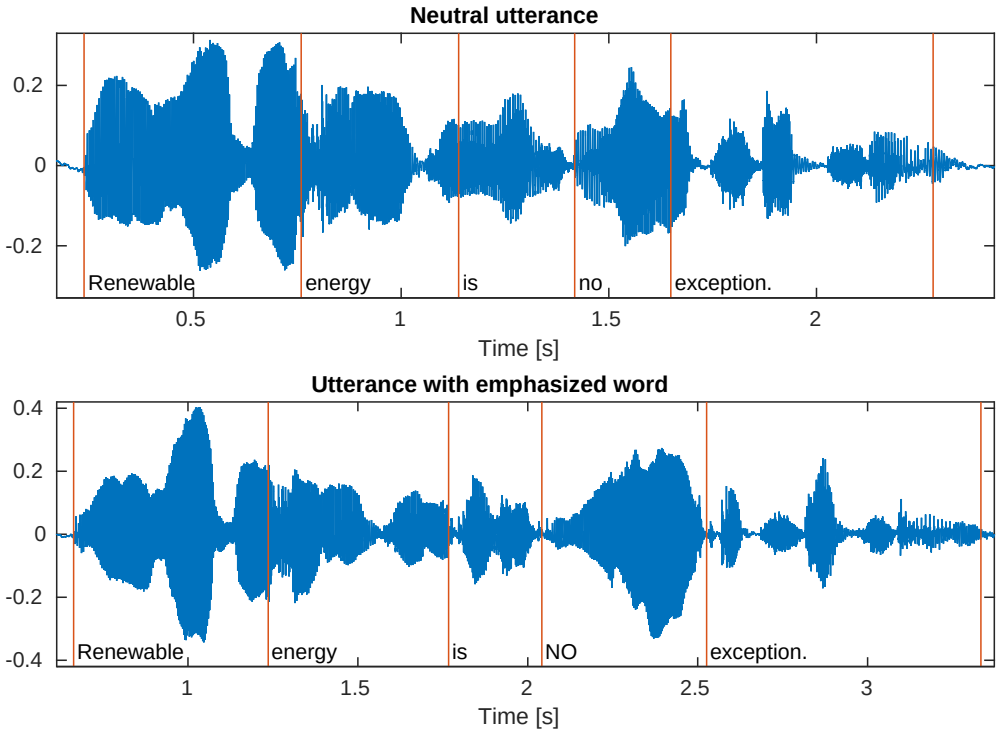



**Figure 1.1:** *Comparison of neutral and partially emphasized utterances*

**Classification** The goal of this work is the prediction of emphasized words. For this purpose, the four derived features need to be combined in a meaningful way.

One approach is to find a function that combines the derived features. Such an approach is taken by [10], where an addition of suitable features is performed. However, addition is not feasible for the features derived in this work, as they vary in range and influence.

Another approach is to find a meaningful combination by machine learning. In [11], for example, multilayer perceptrons are used for the classification of prosodic events. In this work also classifiers were used. In order to find the most suitable classifier architecture, several different types of classifier were trained and evaluated. This process is described in Section 2.4.

# 2 Implementation

In this work the process of emphasis detection is divided into three steps, that is, raw feature extraction, derived feature calculation and classification. Each of these steps is described in detail in this chapter. All implementation was done in MATLAB.

The development of a method for emphasis detection requires a speech signal data set. More information on the data set used in this work is provided in Section 2.1.

## 2.1 Data Set

This project requires a set of audio samples containing emphasized words. Gathering and annotating speech signals is cumbersome. Hence, it was advisable to work with a preexisting corpus. A suitable set for this project was the multilingual SIWIS corpus, which was created as a part of the SIWIS project [6].

When this corpus was recorded, 22 speakers were asked to record 50 sentences each. 25 sentences contain normal prosody and the other 25 are the same sentences with intentional emphasis on a couple of words. Out of the 22 speakers 10 were male and 12 were female. All of the speakers recorded utterances in at least two languages. The corpus contains recordings in German (DE), English (EN), French (FR) and Italian (IT).

The audio signals and their textual representation provided by the multilingual SIWIS corpus do not contain all necessary information for this work. This work's feature extraction also required a phonetic transcription and a segmentation of the example signals. Both were produced with lab internal software of the ETH speech processing group [1] and provided for this work.

The provided phonetic transcription files indicate phrase, word and syllable borders, syllable stress and phonemes. The segmentation files include start and end times of each phoneme. The segmentation software split most of the plosives into a closure and a release part. This allowed for a separation of the two parts when calculating phoneme statistics, which lead to more accurate results.

The phonetic transcription uses symbols of the ETH computer phonetic alphabet (ETHPA). An overview of these symbols is given in C.

While the SIWIS corpus contains samples in four languages, segmentation was only possible for German, English and French. This is why in this work the Italian samples were omitted. Table 2.1 lists the number of available audio samples and Table 2.2 shows a statistical evaluation of the length of these samples.

| | DE | EN | FR | All languages |
|---|---|---|---|---|
| Female speakers | 8 | 9 | 11 | 12 |
| Male speakers | 3 | 7 | 8 | 10 |
| Files with no emphasis | 292 | 402 | 506 | 1200 |
| Files with emphasis | 207 | 334 | 358 | 899 |
| Total files | 499 | 736 | 864 | 2099 |
| Neutral words | 3606 | 5842 | 6985 | 16433 |
| Emphasized words | 262 | 415 | 458 | 1135 |
| Total words | 3868 | 6257 | 7443 | 17568 |

**Table 2.1:** *Data set size*

| | DE | EN | FR | All languages |
|---|---|---|---|---|
| Average sentence length | 3.24 s<br>7.8 words | 2.98 s<br>8.5 words | 2.89 s<br>8.6 words | 3.01 s<br>8.4 words |
| Average emphasis length | 0.81 s<br>1.3 words | 0.77 s<br>1.3 words | 0.75 s<br>1.4 words | 0.77 s<br>1.3 words |
| Minimum sentence length | 0.82 s<br>3 words | 0.94 s<br>4 words | 1.03 s<br>4 words | 0.82 s<br>3 words |
| Minimum emphasis length | 0.22 s<br>1 word | 0.12 s<br>1 word | 0.09 s<br>1 word | 0.09 s<br>1 word |
| Maximum sentence length | 7.16 s<br>17 words | 5.62 s<br>15 words | 7.24 s<br>20 words | 7.24 s<br>20 words |
| Maximum emphasis length | 2.54 s<br>3 words | 2.97 s<br>4 words | 2.84 s<br>5 words | 2.97 s<br>5 words |

**Table 2.2:** *Mean duration of sample speech signals and their emphasized parts*

## 2.2 Primary Features

The four measurable prosodic quantities in a speech signal are fundamental frequency profile, intensity profile and phoneme and pause duration [12]. All of them are influenced by emphasis. Generally speaking, emphasis increases duration, magnifies intensity and results in a higher pitch. Thus, in a first step of this work, these primary features were extracted for each recording in the data set.

In Section 2.2.1 the combined extraction of pause and phoneme duration is presented. Section 2.2.2 contains details on intensity feature extraction. Fundamental frequency extraction is described under Section 2.2.3.

Fundamental frequency was calculated as a function of time. Duration and intensity features were both calculated on a phoneme level. Deriving higher order features from duration and intensity required a statistical analysis of these raw features. Since the data set is rather small, a word based statistic was not possible. Therefore, the statistics were accumulated once for each phoneme separately and thereafter for different phoneme classes. The phoneme classes are shown in Table 2.3.

### 2.2.1 Duration

The segmentation files provided the start and end times of each phoneme of an utterance. The duration of a phoneme was determined by taking the difference between its start and end time as shown in (1).

$$\mathrm{dur}_{\mathrm{phoneme}} = \mathrm{t}^{\mathrm{end}}_{\mathrm{phoneme}} - \mathrm{t}^{\mathrm{start}}_{\mathrm{phoneme}} \tag{1}$$

This calculation was done for all phonemes and a statistic for each phoneme as well as for all phoneme classes was compiled. For the raw feature extraction pauses within a sentence were interpreted as separate phonemes and also included in the statistics. Some of the recordings have a rather long pause at the beginning and end due to the recording circumstances. These start and end pauses were not included in any statistics since they do not belong to the actual utterance.

| | DE | EN | FR |
|---|---|---|---|
| Short vowel | ✓ | ✓ | ✓ |
| Long vowel | ✓ | ✓ | |
| Diphtong | ✓ | ✓ | |
| Nasal vowel | | | ✓ |
| Plosive closure | ✓ | ✓ | ✓ |
| Plosive release | ✓ | ✓ | ✓ |
| Nasal | ✓ | ✓ | ✓ |
| Fricative | ✓ | ✓ | ✓ |
| No class | ✓ | ✓ | ✓ |

**Table 2.3:** *Phoneme classes for German, English and French*

### 2.2.2 Intensity

For each utterance an intensity contour was calculated using the root mean square (rms) value over a window of 30ms with a 5ms shift. The intensity of each phoneme was then determined by taking the maximum of the intensity contour over the duration of the phoneme. Figure 2.1 shows an example of how the intensity values of a phoneme were determined. The red circles indicate the extracted intensity values for each phoneme.

In order to account for intensity differences resulting from varying recording conditions (such as the speaker being closer to or further away from the microphone), the resulting intensity values were normalized by dividing them by the mean rms value of the whole audio signal. The drawback of the normalization is, that it might level out the intensity contributions of emphasized parts.

Based on the extracted phoneme intensity values, a statistic was constructed listing the intensity's mean and variance as well as the number of each phoneme's occurrences. The values in this statistic were used in the calculation of the derived intensity feature.

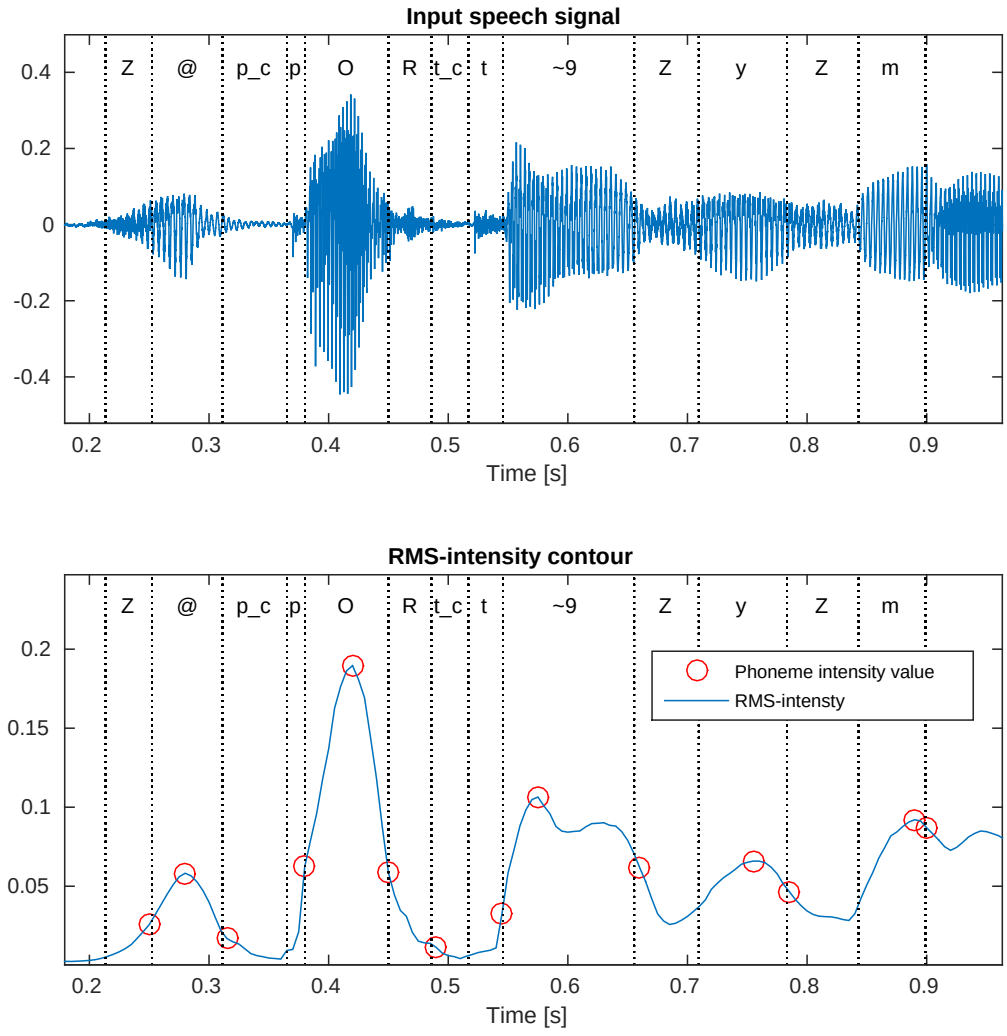



**Figure 2.1:** *Example of an intensity contour with indicated intensity value for each phoneme*

### 2.2.3 Fundamental Frequency

Within the speech processing group of ETH there are several lab internal MATLAB methods suited to extract fundamental frequency from an audio signal. The two considered for this work were `detect_f0_contour.m` and `getfundamentalfreq.m`. The former estimates fundamental frequency by finding the most probable contour in a high-resolution cepstrogram as described in [9]. The latter approximates fundamental frequency using the input signal's cepstrum. Where this fails, the latter method uses autocorrelation, frequency contour prolongation in the spectogram and interpolation. Unvoiced frames are marked as such.

Figure 2.2 shows a comparison of the two fundamental frequency extraction methods. Both extraction methods were applied to the same signal for different window sizes. The comparison shows that `getfundamentalfreq.m` delivers inconsistent results when comparing window sizes of $0.1$ and $0.05$ seconds. Hence, `detect_f0_contour.m` was chosen to extract fundamental frequency contours in this work.

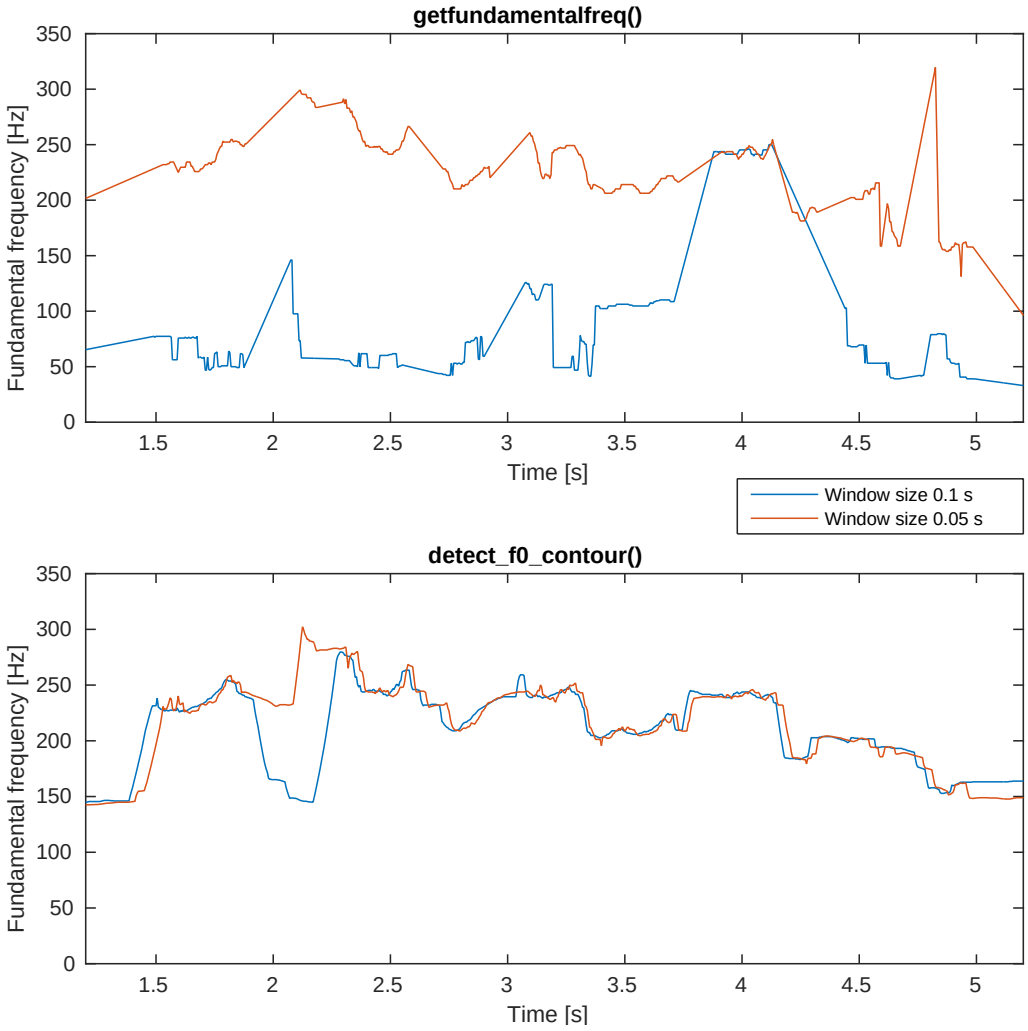



**Figure 2.2:** *Comparison of fundamental frequency extraction methods*

Related to fundamental frequency extraction is voicing information. While some parts of a speech signal are voiced, others may be silence, unvoiced or something in between. Fundamental frequency information is most relevant for frames which belong to voiced parts of a signal. The lab internally provided function `detect_voicing_information.m` classifies frames into the five categories unvoiced, silence, voiced, mixed excitation and irregularly voiced. For further calculations the unvoiced and silence frames were discarded.

## 2.3 Derived Features

Directly applying machine learning methods to the extracted raw features described above would have required very powerful classifiers and an immensely large training data set. Since the size of the given training set was limited, it was necessary to condense the information contained in the raw features. This was done by calculating derived features from the input raw features. Ideally, such derived features contain as much of the raw feature information relevant to emphasis as possible while discarding overhead.

Some of the derived features in this work were inspired by feature extraction methods for prosodic analysis in existing work such as [8] and [10]. For the others, the general approach to finding suitable derived features was to compare plots of raw features of neutral signals with their emphasized counterparts. Possible good features were then implemented and evaluated using Fisher distances. As this method worked best for each raw feature separately, one derived feature was calculated for each of the four measurable prosodic speech signal quantities. Each of these derived features is described in detail in the following sections. There was no combination of different raw features in the calculation of one derived feature. The combination of frequency, intensity, phoneme and pause duration was done by the classifiers in the next step.

Other than containing as much information on emphasis as possible there were several other requirements for the derived features. One requirement stemmed from classifiers needing fixed length input vectors per classification unit. Possible classification units were time frames, phonemes, syllables or words.

Classifying each time frame separately was impractical, because some of the raw features were given on phoneme level (e.g. phoneme duration). Classification per phoneme or syllable was possible. However, if phonemes or syllables had been chosen as classification unit, this would have resulted in the need for another combination or classification step. Given that not all phonemes or syllables of one word may have been classified as emphasized, an additional step would have been needed to decide whether or not the whole word was emphasized. Such an additional step would have made the evaluation more complicated. Hence, in this work words were chosen as classification unit.

With words set as classification unit, each derived feature needed to produce a constant number of values per word. This number did not have to be the same for different features, as long as it was constant within one feature. While any constant number was admissible, one value has proven to be sufficient. Hence, every derived feature resulted in one value per word.

Another requirement for derived feature calculation was the reduction of influences other than emphasis. Prosodic quantities are influenced by many factors. Fundamental frequency, for example, tends to be higher at the beginning of a prosodic phrase and lower towards the end. Such influences, that do not stem from emphasis, had to be excluded from the derived features.

### 2.3.1 Duration

The duration of an emphasized word is inclined to be longer than that of the same neutral word. Thus, as a meaningful indicator of emphasis, the duration of a word needs to be compared to its statistical neutral length. Due to a limited data set, in this work it was not possible to establish word level statistics. Hence, the comparison had to be done on a phoneme level. In search of a word based duration feature, an approach from [10] was explored and applied to the provided data set. The relative word prolongation (rwp) of a word $w$ is defined in (2). $\mathrm{dur}(w)$ is the sum of the duration of each phoneme in the word $w$, and $\sum_{p \in w} \mathrm{stat\_dur}(p)$ is the sum of the statistical duration of each phoneme. The statistical duration of a phoneme is the mean of the durations of all neutral occurrences of that phoneme in the given data set.

$$\mathrm{rwp}(w) = \frac{\mathrm{dur}(w) - \sum_{p \in w} \mathrm{stat\_dur}(p)}{\#\mathrm{phonemes}} \tag{2}$$

The rwp takes into account the statistical duration of a word by summing up the statistical durations of each of its phonemes. Using rwp circumvented the need for word level statistics.

A second approach for deriving a duration feature was inspired by [8]. The idea was to compare the word and its surrounding words to how an "average speaker" would utter the context. This approach resulted in two features per word $\tau_{dur}$ and $\zeta_{dur}$. An extensive explanation and the formulas for calculating these values are given in [8]. However, the obtained Fisher distances for these features were not as high as the ones of the rwp. The comparison of the Fisher distances between the two approaches can be found in Section 3.1.2. The decision on a derived duration feature was made in favor of the rwp.

### 2.3.2 Intensity

An emphasized word typically shows an increased intensity compared to its neutral counterpart. However, not all phonemes' intensity is increased the same. In order to differentiate between *emphasized* and *neutral*, the intensity of each phoneme needs to be compared to its statistical neutral value. [10] used such a comparative feature for detecting emphasized words, which was adapted to this work's purpose. (3) shows the definition of the relative intensity increase (rii) of a word $w$ with its most prominent syllable being $\hat{\sigma}$. $\mathrm{rms}(\hat{\sigma}_{nuc})$ signifies the intensity of the nucleus of a word's most prominent syllable, and $\mathrm{stat\_rms}(\hat{\sigma}_{nuc})$ is the corresponding statistical value. The statistical intensity of a phoneme was the mean of the intensity off all neutral occurences of that phoneme in the given data set.

$$\mathrm{rii}(w) = \mathrm{rms}(\hat{\sigma}_{nuc}) - \mathrm{stat\_rms}(\hat{\sigma}_{nuc}) \tag{3}$$

Only the intensity of the most prominent syllable is used as a discriminative feature. This is based on the assumption that emphasis influences this most prominent syllable's intensity the strongest.

The rii in this work was slightly altered from what [10] proposed. Instead of using the rms values of all phonemes in a syllable, only the syllable nucleus rms values were considered. In experiments, this type of calculation showed better distinction between neutral and emphasized words.

Another idea was again inspired by [8]. The same approach as mentioned in Section 2.3.1 was applied to intensity. Each word resulted in two corresponding features $\tau_{int}$ and $\zeta_{int}$. An extensive explanation and the formulas for calculating these values are given in [8]. The context of a word was compared to how intense an "average speaker" would speak it. However, the Fisher distances for this approach were lower than for the rii. Thus, the rii was chosen as the intensity feature. In Section 3.1.2 a comparison of the Fisher distances between the two approaches is listed.

### 2.3.3 Fundamental Frequency

Emphasized words tend to display increased fundamental frequency. Thus, fundamental frequency is a suitable feature to describe emphasis. However, fundamental frequency is also influenced by other factors. Looking at the fundamental frequency value of one frame, the value will strongly depend on the frame's location in a prosodic phrase and the syllable stress level.

In order to alleviate the influence of prosodic phrase position, it was necessary to look at fundamental frequency deviation rather than absolute fundamental frequency values. Rather than the absolute fundamental frequency contour, one should measure the difference between the fundamental frequency contour and a smoothed version thereof. This required the calculation of a smoothed fundamental frequency contour.

Figure 2.3 shows the two approaches for smoothing the fundamental frequency contour taken into consideration in this work. Generally, fundamental frequency tends to decay over the progress of one prosodic phrase. Hence, one approach was to take a linear regression of the fundamental frequency in each prosodic phrase. This was used in [8]. However, most speakers show some deviations from the general pattern of linear fundamental frequency decay over one prosodic phrase. Using a short-term average as approximation instead helped taking such deviations in fundamental frequency pattern into account. This is why in this work the final fundamental frequency features were calculated with the help of short-term averages, not linear fundamental frequency regression.

The short-term average was calculated by averaging the fundamental frequency values of all frames within a certain window. For the average's calculation silence and unvoiced frames were discarded. It was necessary to find a good value for the size of the window in consideration. For this purpose, the Fisher distance between the fundamental frequency features of emphasized words and neutral words was compared for different window sizes. As can be seen in Figure 2.4, a window size of 3 seconds lead to good results for all three languages. Hence, the final fundamental frequency features were calculated using a short-term average over a window of 3 seconds.

Another influence to be alleviated was syllable stress level. The selected approach for this consists of two steps. Initially, the fundamental frequency feature was calculated for each syllable. Subsequently, the highest syllable fundamental frequency increase was assigned to the entire word. This is also in accordance with the tendency of emphasis to increase fundamental frequency most strongly in dominant syllables.

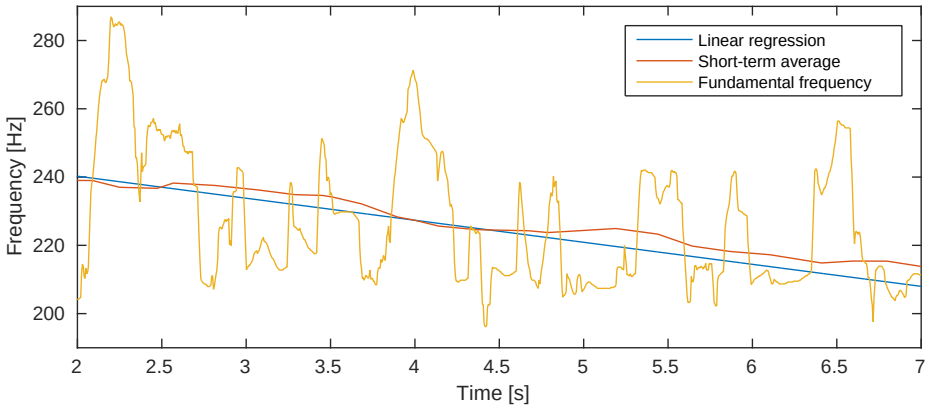



**Figure 2.3:** *Fundamental frequency profile approximated with linear regression and short-term average*

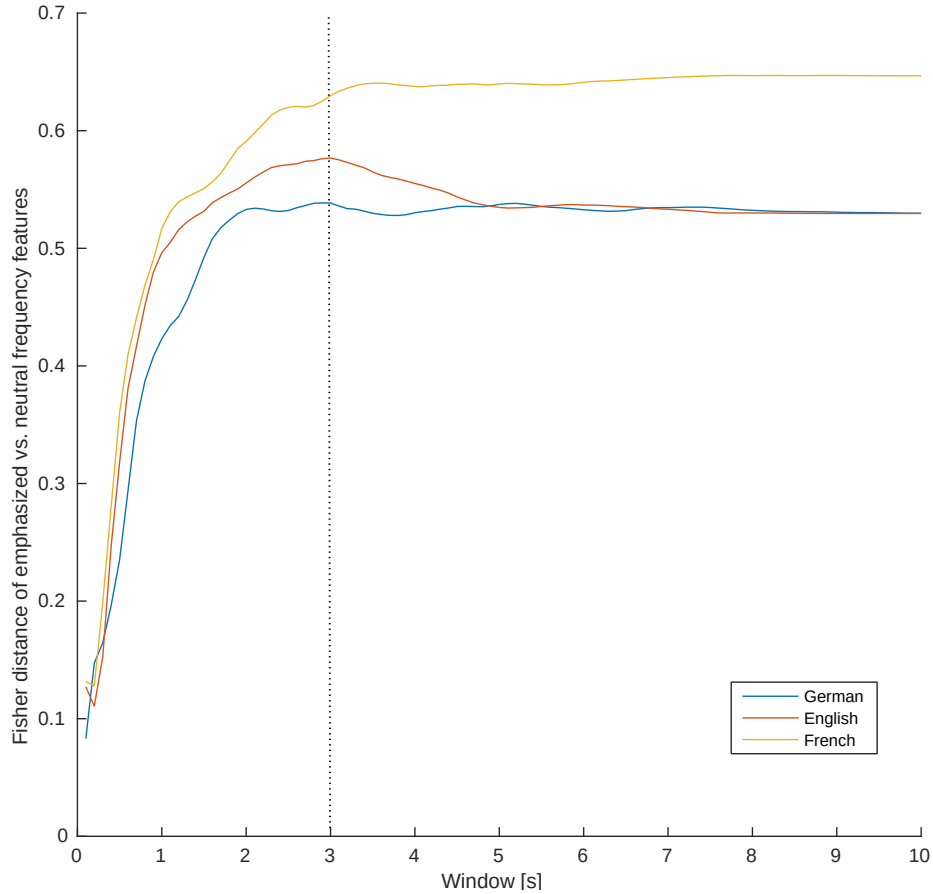



**Figure 2.4:** *Fisher distances between fundamental frequency features of emphasized and neutral words as a function of short-term average window size*

Originally, the fundamental frequency increase was calculated per syllable and then averaged over a word to result in one value per word. Because this approach did not take into consideration dominant syllables, it was performing worse than the approach described right above. A comparison of the two approaches' performances is presented under Section 3.1.2.

The alleviation of syllable stress level demanded the calculation of one derived fundamental frequency feature value per syllable. It had to be decided, which part of each syllable's fundamental frequency contour was to be compared to the fundamental frequency short-term average. Generally, the syllable nucleus carries more valuable information on fundamental frequency than the syllable transitions. Hence, for each syllable the average fundamental frequency of its nucleus was calculated. Figure 2.5 shows the calculated nucleus fundamental frequency averages in the partial sentence "I look forward to hearing". The light blue shaded nuclei were detected automatically using the assumption that all vowels belong to a syllable nucleus.

Once the short-term fundamental frequency average at each syllable center and each syllable's nucleus fundamental frequency average were determined, the syllable's fundamental frequency feature was calculated as the difference of those two values.

Fundamental frequency of a speech signal depends on the speaker's gender. This is also true for the variance of a signal's fundamental frequency. Depending on the type of derived fundamental frequency this features may be gender-dependent. In that case a normalization would be necessary to abolish the gender differences. However, for the fundamental frequency features derived as described above, the gender difference was negligible. Figure 2.6 shows the fundamental frequency feature distribution's cumulative density of neutral and emphasized words for both genders. As seen, the features depend a lot more on emphasis than they do on gender.

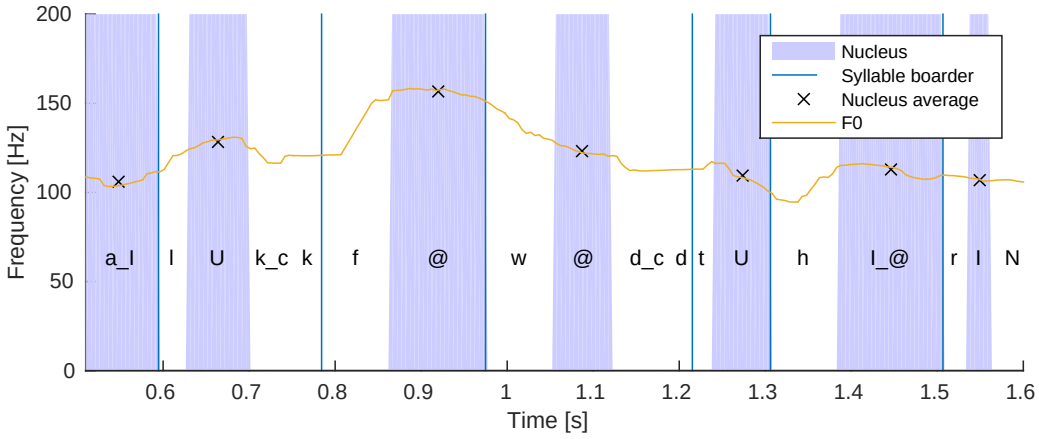



**Figure 2.5:** *Fundamental frequency averages of detected nuclei in the partial sentence "I look forward to hearing"*

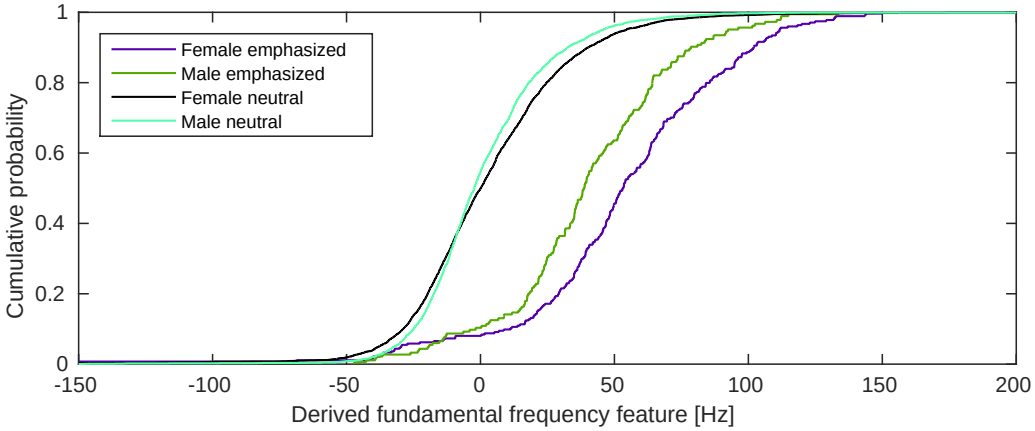



**Figure 2.6:** *Gender dependency of fundamental frequency feature visualized as CDF*

### 2.3.4 Pause Duration

Table 2.4 gives an overview of the average duration of pauses and plosive closures depending on their location in a sentence. Table 2.5 lists the probability with which a segment is a pause. This statistical analysis of the audio samples in the multilingual SIWIS corpus reveals that emphasis leads to both longer and more frequent pauses. The findings correspond to what one would intuitively assume about the influence of emphasis on pauses.

The difficulty in deriving a fixed length pause feature for each word, lies in the fact that not every word is surrounded by pauses or contains any of them. As seen in Table 2.5, pauses within words are a very rare occurrence. This is why they were not included in the calculation of pause features.

After discarding pauses within words, pause information was transformed into fixed length features by taking the duration of pauses in front and behind the word as feature values. If there was no pause, the value was set to zero. This resulted in a feature of length two for every word. This was one of numerous possible pause features considered in this work.

| | Mean pause duration | | | Mean closure duration | | |
|---|---|---|---|---|---|---|
| | DE | EN | FR | DE | EN | FR |
| Within emphasized words | 185.5 ms | 135.9 ms | 186.6 ms | 54.2 ms | 59.0 ms | 74.1 ms |
| Between emph. words | 94.3 ms | 73.7 ms | 129.8 ms | 52.9 ms | 71.1 ms | 74.3 ms |
| Right before emph. words | 138.1 ms | 156.5 ms | 177.4 ms | 64.3 ms | 73.8 ms | 83.8 ms |
| Right behind emph. words | 169.5 ms | 185.1 ms | 183.6 ms | 63.1 ms | 55.2 ms | 70.8 ms |
| Within neutral words | 129.2 ms | 117.4 ms | 123.7 ms | 48.9 ms | 50.3 ms | 56.5 ms |
| Between neutral words | 138.7 ms | 131.3 ms | 138.9 ms | 52.8 ms | 50.6 ms | 56.1 ms |

**Table 2.4:** *Average pause and plosive closure duration for different sentence locations*

| | Probabilty of character being a pause | | |
|---|---|---|---|
| | DE | EN | FR |
| Within emphasized words | 0.63% | 1.03% | 1.40% |
| Between emphasized words | 8.00% | 12.33% | 16.28% |
| Right before emphasized words | 23.70% | 23.39% | 24.19% |
| Right behind emphasized words | 33.02% | 43.71% | 41.13% |
| Within neutral words | 0.26% | 0.42% | 0.53% |
| Between neutral words | 7.28% | 7.64% | 7.04% |

**Table 2.5:** *Probability of pause occurrence for different sentence locations*

Other versions of the feature took into account that pauses are not detected in front of plosives. Pauses before plosives are always listed as part of the plosive closure. Hence, for the second version of the feature, both plosive closures and pauses were set as feature values. Absence of plosive or pause still resulted in a zero feature entry.

Counting plosive closure and pauses equally may put too much weight on plosive closures. Another version of the feature did not use the entire plosive closure, but only added plosive closure elongation as a feature. The elongation was calculated in two different ways. For one version the elongation was equal to the measured plosive closure minus the general mean plosive closure for that language. For another version the elongation was calculated as the difference between measured closure and the mean duration of the closure for the given phoneme for that language.

In two other versions of the feature, duration information was discarded. For one of them, both pauses and plosive closures were translated into a feature value of one. Their absence was translated into zero. For another, only pauses were counted as ones, everything else as zero.

All the pause feature versions above delivered two values per word. One way of reducing these two values into one was to add them up. This was done for all 6 previously described feature versions.

Another way of reducing the two values to one, was by discarding one of them. Tests have shown that the values based on pauses after a word contain more information on whether the word is emphasized than the ones before.

To find out which feature version performs best, Fisher distances were calculated. An overview of the Fisher distances of the best performing features is given in the results section of this work (Chapter 3). For all three languages, the best performing feature was the added up combination of pause and full closure duration.

## 2.4 Classification

Nowadays an abundance of ready-to-use implementations of classifiers exist. This work's feature extraction was implemented in MATLAB. Thus, it suggested itself to use the MATLAB Statistics and Machine Learning Toolbox [4] for classification. All classifier implementations used in this work, with the exception of hidden Markov models, stem from this toolbox.

A total of four different classifier architectures were evaluated in this work. Each of them is described briefly below. Furthermore, Section 2.4.1 gives an overview of the samples used for classifier training and Section 2.4.6 shows the approach for parameter optimization, which all four classifier architectures have in common.

### 2.4.1 Classification Data Samples

The calculation of derived features described above resulted in four values per word, one for each feature. This allowed to build a data set which was suitable to train classifiers for emphasized word detection. Each word equated to one sample, consisting of a feature vector and a label. A sample's feature vector contained the four derived features. The label of the sample indicated whether it was emphasized or neutral.

For each language one data set as described above was built. These data sets were of the right format to train and evaluate classifiers.

The multilingual SIWIS corpus contains a lot more neutral words than emphasized ones (cf. Table 2.1). The resulting imbalance amongst the two classes of samples may reduce performance in certain classifiers. For that purpose, balanced data sets were generated in addition to the full data set described above. The balanced sets contained all emphasized samples of a given language and an equal number of randomly chosen neutral samples.

For the evaluation of hidden Markov models, support vector machines and neural networks, 10-fold cross-validation was performed. This was to avoid any loss of generality, which testing a classifier with its own training data would entail. When performing 10-fold cross-validation on the balanced set, it was ensured, that the 1:1 ratio between neutral and emphasized samples was kept for all the folds. Random forests were evaluated using the out-of-bag classification error.

For 10-fold cross-validation with balanced training sets, the number of available training samples was given as $\frac{2 \cdot 9}{10}$ times the number of one language's emphasized samples. For 10-fold cross-validation with unbalanced training sets, the number of available training samples was given as $\frac{9}{10}$ times the total number of a language's samples. Table 2.6 contains an overview of the number of available training and testing samples for the training and testing of each classifier in 10-fold cross-validation. Values are stated for both, training with a balanced data set and training with the full set.

### 2.4.2 Support Vector Machines

Training a binary support vector machine equates to finding the ideal separating hyperplane between training samples of two classes in a feature vector space. A more detailed description of support vector machines can be found under [5].

| | Training samples | | | Testing samples | | |
|---|---|---|---|---|---|---|
| | DE | EN | FR | DE | EN | FR |
| Balanced | 471 | 747 | 824 | 3397 | 5510 | 6619 |
| Unbalanced | 3481 | 5631 | 6698 | 387 | 626 | 745 |

**Table 2.6:** *Number of available training and testing samples for balanced and unbalanced training sets*

The support vector machine implementation provided as part of the MATLAB Statistics and Machine Learning Toolbox [4] allows for the adjustment of multiple parameters. Below, a short overview of the most important ones with regard to this work is given.

**Standardization** It is commonly acknowledged as good practice to standardize features for machine learning. To ensure that standardization is not contra productive, some tests are run without it.

**Cost** Training a support vector machine with an unbalanced data set may produce a classifier that is biased towards the more frequent class. There are two solutions of this problem. One is to artificially balance the data set by leaving out part of the more frequent class's samples (here: neutral samples). The other is to assign a higher cost of misclassification to samples of the less frequent class. In this work, both methods have been assessed.

**Kernel** The default kernel of support vector machines is linear. This results in the construction of a hyperplane between the two classes which are to be separated. However, some classes need a non-linear separation. This is achieved by using different kernels. The derived features in this work were not linearly separable. Hence non linear kernels had to be used.

### 2.4.3 Random Forests

A random forest is an ensemble learning method which constructs a predefined number of decision trees during training and classifies a new sample according to a majority vote of the individual trees. More information on random forests and how they can be implemented in MATLAB can be found under [3]. For the training of a random forest, bagging (bootstrap aggregating) was applied. During bagging, for each decision tree that is being built, a new training set of the same size $D$ as the original training set is constructed by sampling the original training set $D$ times with replacement. The training samples that have not been selected for the current tree are used for testing. This method makes cross-validation for random forests redundant.

The MATLAB Statistics and Machine Learning Toolbox [4] provides a framework for random forests with several possible parameters to modify. The most interesting adjustable parameters for this work are shortly explained below.

**Number of trees** According to [7] random forests do not overfit the data. Hence, an arbitrary high number of trees can be grown. For this work the number of trees was set to 2000.

**Cost** In order to get a less biased classifier in case of an unbalanced training set, a cost matrix can be defined. With this matrix the training samples of the less represented class can be assigned a higher cost of misclassification.

**Minimum leaf size** The minimum leaf size can prevent overfitting for individual decision trees. It determines the minimum number of observations of for each tree leaf. A larger minimum leaf size reduces the depth of a tree. For random forests it is used as a mean to speed up training.

**Decision split variables** At each decision split of a tree a certain number of feature variables are chosen at random to optimize the split. Increasing the number of variables to choose from increases both the correlation between two trees in the forest and the strength of each individual tree. Decreasing it reduces both the correlation and the strength. Thus, there is a tradeoff between correlated but stronger trees and uncorrelated but weaker trees.

Because random forest training was calculation intensive, optimization was done in two parts. First, the minimum leaf size and number of decision split variables were optimized. The criterion for this was the $F_1$ score. In a second step, the random forest with the best minimum leaf size and number of decision split variables was optimized over different cost matrices. The validity of this two step optimization is based on the assumption that the cost matrix's influence and the influence of the minimum leaf size and the number of decision split variables are independent regarding performance. While such an assumption was necessary, its validity may be limited. For future work, a more comprehensive optimization should be taken into consideration.

### 2.4.4 Hidden Markov Models

A two-state hidden Markov model (HMM) can be used for binary classification. For that purpose, each state is set to represent one of the two classes. The input features are taken as HMM observations. As the features found in Section 2.3 are continuous, the observations of the HMM used for emphasis detection need to be of continuous densities. The HMM used for this task is therefore a two state continuous density hidden Markov model (CDHMM). Figure 2.7 depicts a graphical representation of the utilized CDHMM.

The observations *duration* (dur), *intensity* (int), *pause duration* (pdur) and *fundamental frequency* (freq) are continuous and are characterized by a Gaussian mixture distribution $b_j$ for each state $S_j \in \{\textit{neutral}, \textit{emphasized}\}$:

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{M} c_{jk} b_{jk}(\boldsymbol{x}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) \tag{4}$$

$\boldsymbol{x}$ is the input feature vector, $M$ is the number of mixture components, $c_{jk}$ is the weighting factor, $\boldsymbol{\mu}_{jk}$ the mean vector and $\boldsymbol{\Sigma}_{jk}$ the covariance matrix of the $jk^{\text{th}}$ mixture component. For calculation purposes the covariance matrix $\boldsymbol{\Sigma}$ was assumed to be diagonal. Such an assumption neglects the correlation between the elements of the feature vector. The feasibility of this simplification should be evaluated in future work.

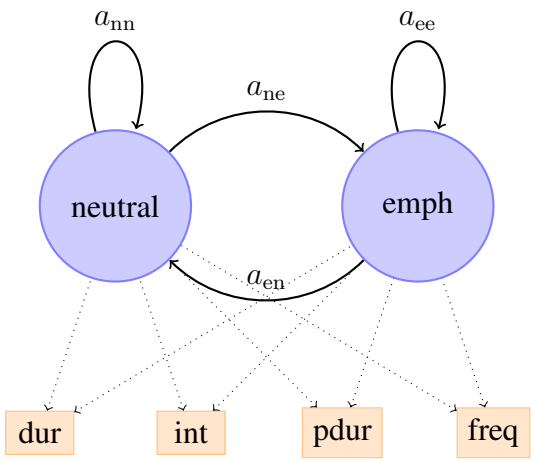



**Figure 2.7:** *Representation of a suitable CDHMM for the purpose of distinguishing between neutral and emphasized words*

The CDHMM was trained by estimating the state transition probabilities and the observation probabilities. The parameters that needed to be trained are the components of the transition matrix $A$, the weighting factors $c_{jk}$, the mean vectors $\boldsymbol{\mu}_{jk}$ and the covariance matrices $\boldsymbol{\Sigma}_{jk}$. The training was conducted in two steps for each cross-validation fold. First, all the transitions from a neutral/emphasized word to a neutral/emphasized word in all the sentences in the current fold were counted and divided by the total number of transitions. From these pseudo probabilities the transmission matrix $A$ was constructed. Then, the Gaussian mixture distribution was estimated for each of the two states and the $c_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ were extracted.

Since an HMM is a temporal model, the input data's sequence is essential. Therefore, in both the training and test data the sentence structure was maintained. This means, that words within a sentence were kept in their order rather than randomly mixed with other words. Only whole sentences were rearranged during cross-validation.

For classification the Viterbi algorithm was applied. This produced an optimal state sequence for the test data of the current cross-validation fold. This state sequence could then be interpreted as the prediction labels. With a small adaptation the Viterbi algorithm provided by the ETH speech processing group [1] also returns a posterior probability for each output. These posterior probabilities can be used as confidence scores needed in ROC plots.

In order to optimize the CDHMM, the performance with different numbers of mixture components $M$ for the observation probabilities was evaluated.

### 2.4.5 Neural Networks

Neural networks are a machine learning tool inspired by biological nervous systems. The type of neural network used in this work is called multilayer perceptron (MLP). A detailed description of such multilayer perceptrons can be found in [12].

The neural network implementation used in this work is called `feedforwardnet` [2]. It is part of the MATLAB Statistics and Machine Learning Toolbox [4]. Another potentially suitable implementation would have been `patternnet` from the same toolbox. However, pattern nets only return labels and no confidence scores. This renders the meaningful generation of ROC-plots impossible.

After choosing a suitable neural network implementation there were several aspects that needed to be taken into consideration. Each of them is described below.

### Training algorithm

The MATLAB feed forward net's default training algorithm is called Levenberg-Marquardt. For this work it was not necessary to change this training algorithm. Both speed and training performance were sufficient. No problems with memory requirements occurred.

### Normalization

The MATLAB feed forward net implementation does not offer a built in standardization option. However, normalization of input features may be beneficial. All neural networks in this work have been trained twice, once with the original input features and once with normalized features. If $\mathcal{F}$ is a $n \times 4$ Matrix containing the input features of $n$ samples, then the normalization of the $m^{\text{th}}$ column was calculated as shown in (5). An ideal neural network input feature has zero mean. Hence, the normalization includes mean subtraction. Taking arctan ensures that the normalized features lie within a finite range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. However, using this normalization did not result in the expected improvements. In most cases, using no normalization yielded better performance. For future optimization, different normalization methods should be taken into consideration.

$$\mathcal{F}(:, m) = \arctan\left(\frac{\mathcal{F}(:, m) - \text{mean}(\mathcal{F}(:, m))}{\text{var}(\mathcal{F}(:, m))}\right) \tag{5}$$

### Hidden layer size

Figure 2.8 shows the graphical representation of a standard configuration MATLAB feed forward net. It contains one hidden layer with ten neurons. During this work also the classification performance of other configurations was evaluated. Training neural networks with a large number of neurons requires a large number of training samples. According to [12] the number of training samples should be at least ten times the number of weights in the network. The number of weights in a network with $K$ hidden layers can be calculated using (6), where $n_i$ is the number of hidden neurons in the $i^{\text{th}}$ hidden layer, $n_0$ the number of neurons in the input layer (this is also the dimension of the feature vectors, which is four in this work) and $n_{K+1}$ is the number of neurons in the output layer (which is always one in this work).

$$\#\text{weights} = \sum_{i=1}^{K+1} n_{i-1} \cdot n_i \tag{6}$$

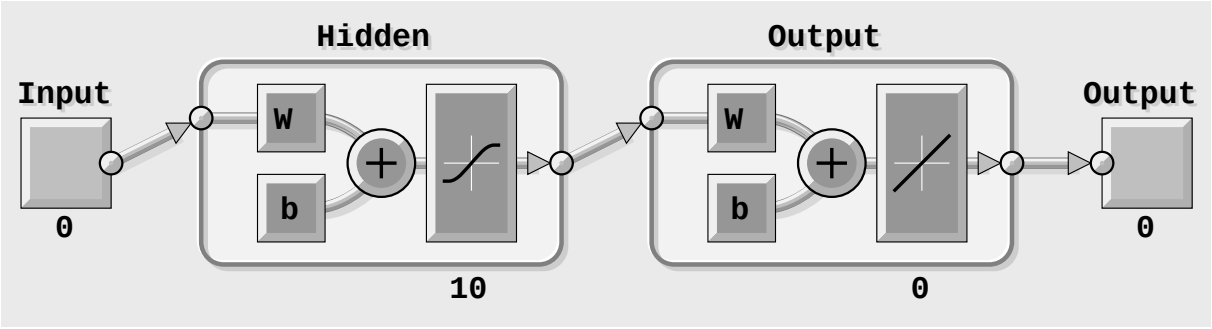



**Figure 2.8:** *Standard configuration of a MATLAB feed forward net*

Starting from the numbers in Table 2.6, the maximal sensible dimensions for hidden layers were calculated. To limit the number of possible network configurations, for every fixed number of hidden layers $K$, the number of neurons in each layer was assumed to be the same ($\nu$). It is possible, that the best performing networks for emphasis classification may not be of this configuration. However, in training, weights of non-essential nodes may tend to become very small and can be pruned out. Hence, the assumption was made, that for any potentially optimal network with variable hidden layer size, there exists a network of comparable size with a predefined number of neurons in each hidden layer, which performs similarly well.

Constraining the configuration possibility as described above facilitates the calculation of the maximal number of neurons in each layer as a function of available training samples ($\#\text{samples}$). This is done by combining (6) with the guideline that $\#\text{weights} < \frac{\#\text{samples}}{10}$ and setting $n_i = \nu$ for all $i \notin \{0, K+1\}$. For a configuration with one hidden layer, a guideline for the maximal number of neurons in this layer was calculated using (7). For configurations with more than one hidden layer, (8) was used to calculate the number of neurons in each hidden layer.

$$n_1 = \frac{\#\text{samples}}{10(n_0 + n_2)} \tag{7}$$

$$\nu = \frac{-(n_0 + n_{K+1}) \pm \sqrt{(n_0 + n_{K+1})^2 + 4(K-1)\frac{\#\text{samples}}{10}}}{2(K-1)} \tag{8}$$

Table 2.7 contains estimates of the highest number of neurons in each hidden layer, which results in a network that can be sensibly trained with the given number of training samples. Such an upper bound was useful, as it limited the range of possible values in parameter optimization.

| # hidden layers | Balanced | | | Full set | | |
|---|---|---|---|---|---|---|
| | DE | EN | FR | DE | EN | FR |
| 1 | 9 | 14 | 16 | 69 | 112 | 133 |
| 2 | 4 | 6 | 6 | 16 | 21 | 23 |
| 3 | 3 | 4 | 5 | 12 | 15 | 17 |
| 4 | 3 | 4 | 4 | 9 | 12 | 14 |

**Table 2.7:** *Maximum number of neurons per hidden layer for given number of training samples*

### 2.4.6 Parameter Optimization

For each of the classifier architectures described in this section, there was a set of parameters that needed to be optimized. For this purpose, multiple instances of the same classifier were trained using different parameter combinations. To identify the most suitable parameter combinations, the performance of the resulting classifiers had to be compared. As the number of parameter combinations is numerous, comparing the resulting classifiers needed to be done automatically. This required a simple performance score for each classifier, the chosen score is called $F_1$ score. This performance measure is described more closely under Section 3.2.1.

For some architectures, classifier performance varied noticeably, even for constant parameters. This can be explained by the changeability of initial weights and the random assignment of data samples into ten folds. To alleviate these fluctuations each classifier was trained with the same parameters multiple times and the resulting performance scores were averaged.

After the best performing configurations were found, it needed to be ensured, that the resulting classifiers had learned to recognize the characteristics of emphasis and not the training samples. This is especially critical for large configurations. Such undesirable classifiers can be detected by comparing their performance on the data they were trained with to their testing data classification performance.

# 3 Evaluation and Results

This chapter describes the experiments used to evaluate feature extraction and classification. It also gives an overview of the results of these experiments. As feature extraction and classification are clearly separated steps in this work's approach, they were also evaluated separately. Nevertheless, it is evident that the quality of the feature extraction strongly influences classification performance.

The evaluation of the feature extraction is described more closely in Section 3.1. Classifier training and evaluation is described in Section 3.2.

## 3.1 Feature Extraction

### 3.1.1 Evaluation Methods

In this work the feature selection and classification were decoupled. Hence, a classifier-independent metric to rate the obtained features was needed. Below, two evaluation criteria are introduced.

**Fisher distance** Fisher distances provide a measure of dissimilarity between two distributions. This measure can be used to rate features without depending on any form of classifier. (9) shows the calculation of the Fisher distance for a feature $\gamma$.

$$\mathrm{F}_{\mathrm{dist}}(\vec{\gamma}) = \frac{\left(mean(\vec{\gamma}_{neutral}) - mean(\vec{\gamma}_{emph})\right)^2}{var(\vec{\gamma}_{neutral}) + var(\vec{\gamma}_{emph})} \tag{9}$$

Fisher distances were used throughout this work. At first they were used to get a sense of the validity of the raw features and thereafter to optimize and rate the derived features.

**Histograms** In this work, histograms were used as a tool to illustrate the distributions of the raw and derived features. When plotted in the same figure, the dissimilarity of the features for neutral and emphasized signal parts could often be seen from eye. Comparison of histograms also allowed for the detection of non-expressive features, which were thereafter discarded.

**Cumulative density function** Histograms are not ideal for the comparison of feature distributions containing a strongly dominating value. In such cases cumulative density functions (CDF) were used for illustration purposes instead.

### 3.1.2 Results

This section summarizes the feature evaluation results of this work. The results of raw feature extraction are presented first. The evaluation of the derived features can be found in the second paragraph.

The presented Fisher distances are always calculated between a feature's neutral occurences and its emphasized occurences.

### Raw Features

Table 3.1 shows the Fisher distances of the three raw features in each language. Figure 3.1 illustrates the histograms of the extracted raw features for German. The histograms for the other languages look comparable and are placed in Appendix A. Figure 3.2 shows the raw feature histograms for all three languages combined.

For the raw duration and intensity features phoneme level statistics were compiled. This allowed to calculate a Fisher distance for each phoneme.

The maximum and minimum Fisher distances based on all the individual phonemes for the raw duration and intensity features are listed in Table 3.2. Due to the limited size of the data set, some phonemes did not occur often enough for their statistical analysis to be significant. Such rare phonemes tend to produce high Fisher distances. However, these high Fisher distances do not actually stem from significant differences between emphasized and neutral speech. They are a result of a too small set of samples. Hence, a Fisher distance's validity should always be confirmed by consulting the corresponding histogram. The cases where phonemes were underrepresented were not included in the calculations for Table 3.2.

Comparing Tables 3.1 and 3.2 reveals, that in some cases Fisher distances for individual phonemes significantly outperform the raw features' Fisher distances. However, for other phonemes the opposite is true. Since not every word consists of well performing phonemes, it is not advisable to use duration and intensity as a phoneme based feature.

Based on the separate phoneme level statistics, phoneme class statistics for the classes in Table 2.3 were compiled. The corresponding Fisher distances are summarized in Table 3.3. The values in brackets are distorted Fisher distances due to a too small number of emphasized samples in the respective class. These values should be ignored.

Only in two cases (intensity of a plosive release in EN and FR) is the Fisher distance for a phoneme class remarkably higher than that of the corresponding raw feature. In all other cases, the Fisher distance is only slightly higher or even lower than that of the raw features. It is thus not advisable to use phoneme class based duration or intensity features for the classification of emphasized words.

| | Fisher distance | | |
|---|---|---|---|
| | DE | EN | FR |
| Duration | 0.117 | 0.113 | 0.162 |
| Intensity | 0.128 | 0.124 | 0.136 |
| Fundamental frequency | 0.095 | 0.058 | 0.122 |

**Table 3.1:** *Fisher distances of the three raw features*

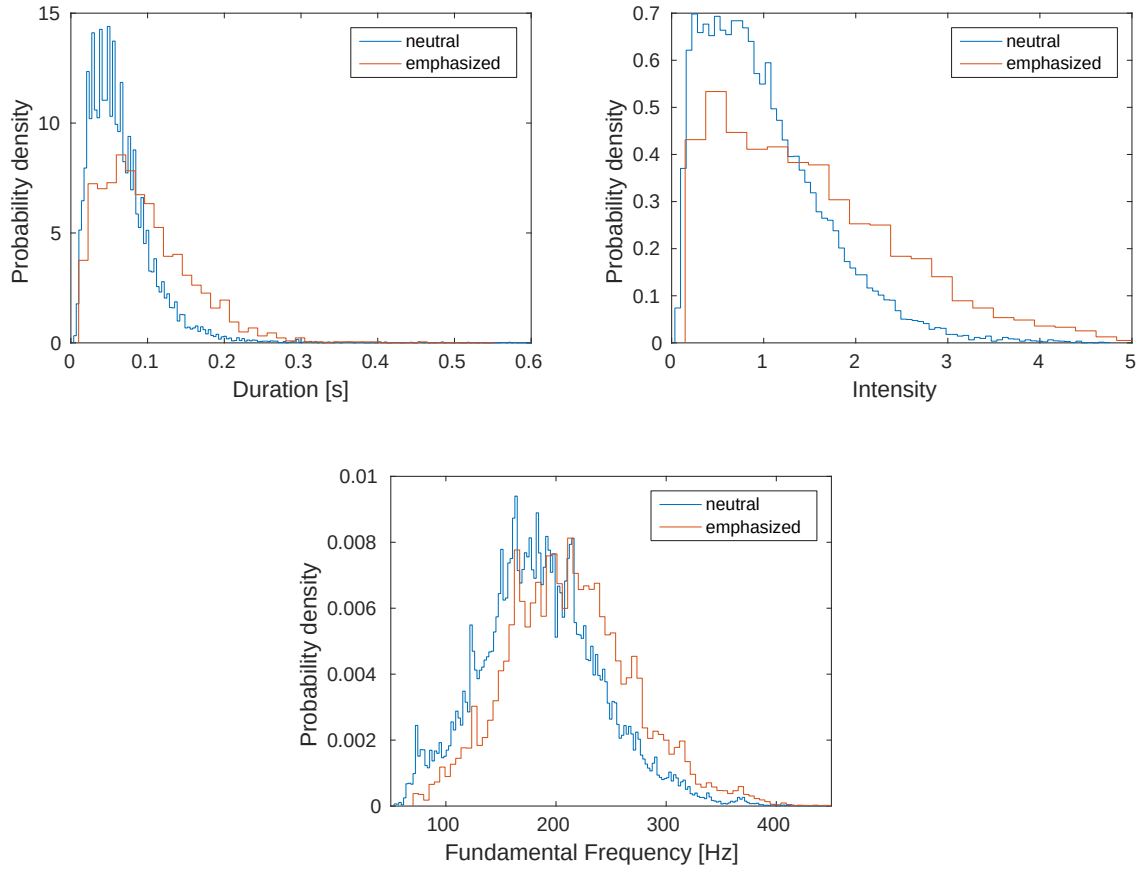



**Figure 3.1:** *Histograms showing the estimated probability distributions of the three raw features of neutral and emphasized words for German*

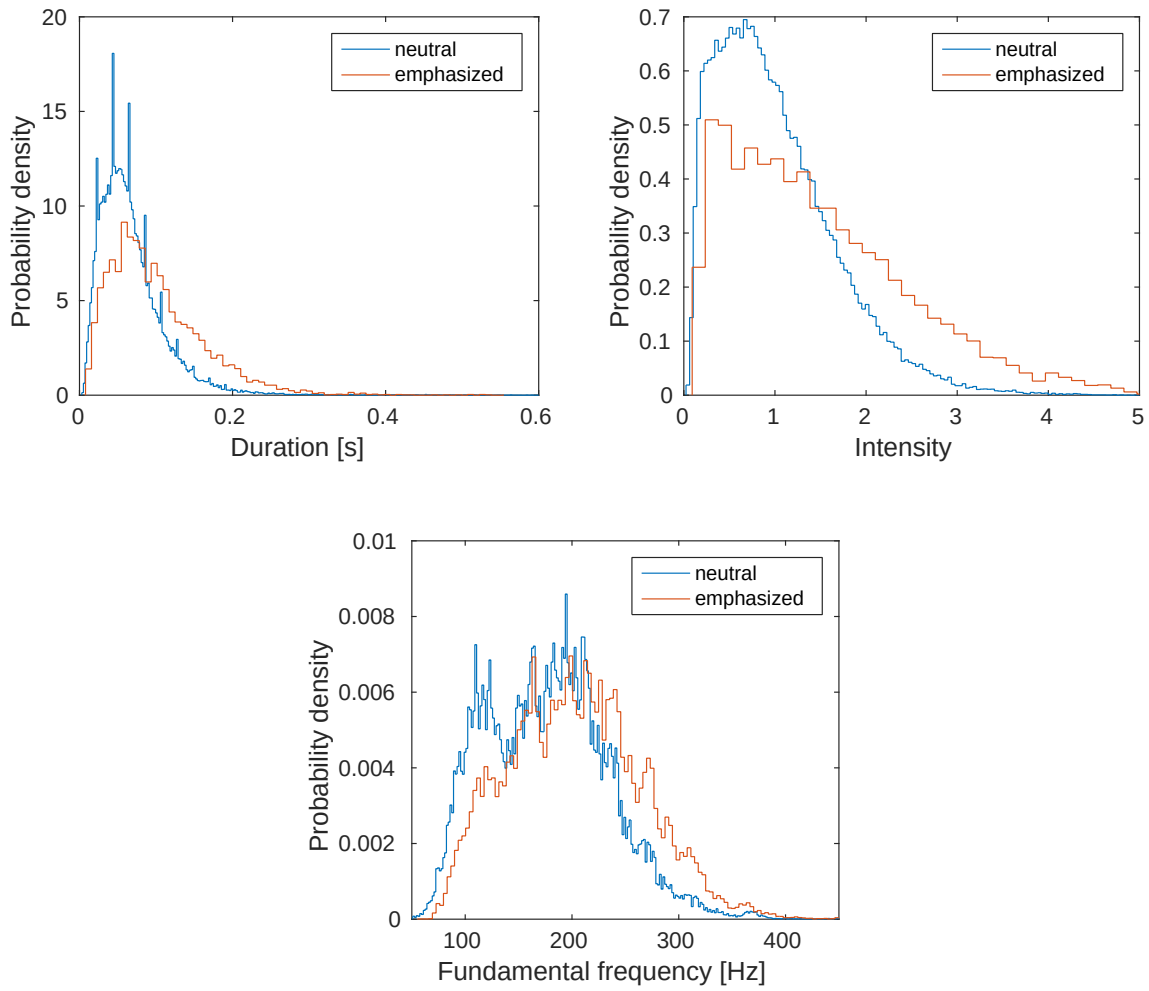



**Figure 3.2:** *Histograms showing the estimated probability distributions of the three raw features of neutral and emphasized words for all languages*

| | Fisher distance | | | | | |
|---|---|---|---|---|---|---|
| | DE | Phoneme | EN | Phoneme | FR | Phoneme |
| Duration maximum | 0.313 | r | 0.324 | v | 0.404 | d_c |
| Duration minimum | 0.006 | n | $1.7 \cdot 10^{-4}$ | Q | $5.0 \cdot 10^{-6}$ | a |
| Intensity maximum | 0.637 | t | 0.921 | t | 1.531 | t |
| Intensity minimum | $3.3 \cdot 10^{-6}$ | l | 0.014 | n | $1.8 \cdot 10^{-4}$ | l |

**Table 3.2:** *Minimum and maximum Fisher distances of raw duration and intensity features based on individual phonemes*

| | Fisher distance | | | | | |
|---|---|---|---|---|---|---|
| | Duration | | | Intensity | | |
| | DE | EN | FR | DE | EN | FR |
| Short vowel | 0.005 | 0.055 | 0.009 | 0.068 | 0.003 | 0.085 |
| Long vowel | 0.125 | 0.010 | - | 0.172 | 0.056 | - |
| Diphtong | (1.829) | 0.129 | - | (0.159) | (0.038) | - |
| Nasal vowel | - | - | 0.007 | - | - | 0.024 |
| Plosive closure | 0.140 | 0.164 | 0.160 | 0.147 | 0.241 | 0.046 |
| Plosive release | 0.254 | 0.144 | 0.177 | 0.394 | 0.874 | 0.647 |
| Nasal | 0.004 | 0.166 | 0.098 | 0.031 | 0.001 | 0.159 |
| Fricative | 0.051 | 0.139 | 0.049 | 0.043 | 0.096 | 0.0172 |

**Table 3.3:** *Fisher distances of raw intensity and duration features based on phoneme classes*

### Derived Features

In this work's progress a multitude of possible derived features was evaluated. Table 3.4 shows the Fisher distances of the derived features which were eventually used in the classification step of this work. Values were calculated for each language separately as well as for all languages combined.

In addition to the Fisher distances, histograms of the derived features were assembled. Figure 3.3 shows the German derived features' histograms. The histograms for the other languages are presented in Appendix A. They show great similarity to the German histograms. The histograms of the four derived features over all languages can be found in Figure 3.4.

For all languages the pause duration feature distribution is less clearly separable than those of the other features. This is why Figure 3.5 contains an alternative illustration of this feature, namely cumulative density plots.

The Figures 3.4 and 3.5 only show the distributions of the best features. These are the final features that were also used for classification. Below, the Fisher distances of these final derived features are compared to the Fisher distances of other features derived from the same raw feature input.

The second approach for deriving both the duration and intensity feature resulted in two features $\tau$ and $\zeta$ per word. In Table 3.5 the $\tau_{dur}$ and $\zeta_{dur}$ features are compared to the chosen duration feature. Table 3.6 presents a comparison of the $\tau_{int}$ and $\zeta_{int}$ to the chosen intensity feature.

In Table 3.7 the final fundamental frequency feature is compared to another version, where the word unit feature was calculated as the mean, rather than the maximum, of this word's syllable features.

Table 3.8 contains a comparison of the six best performing derived pause duration features. All of the best pause duration features use a combination of both closure and pauses. They all use continuous values rather than booleans. They differ in the type of closure duration adjustment (none, phoneme based or plosive class based) and in how the two values, one from pauses and closures before the word and the other from pauses and closure after it, were combined into one. For some of them the two values were added and for others the values before the word were disregarded.

| | Fisher distance | | | |
|---|---|---|---|---|
| | DE | EN | FR | All languages |
| Duration feature | 0.651 | 0.552 | 0.618 | 0.561 |
| Intensity feature | 0.935 | 1.107 | 0.523 | 0.790 |
| Fundamental frequency feature | 0.896 | 0.799 | 0.779 | 0.811 |
| Pause duration feature | 0.168 | 0.230 | 0.360 | 0.258 |

**Table 3.4:** *Fisher distances of the four derived features*

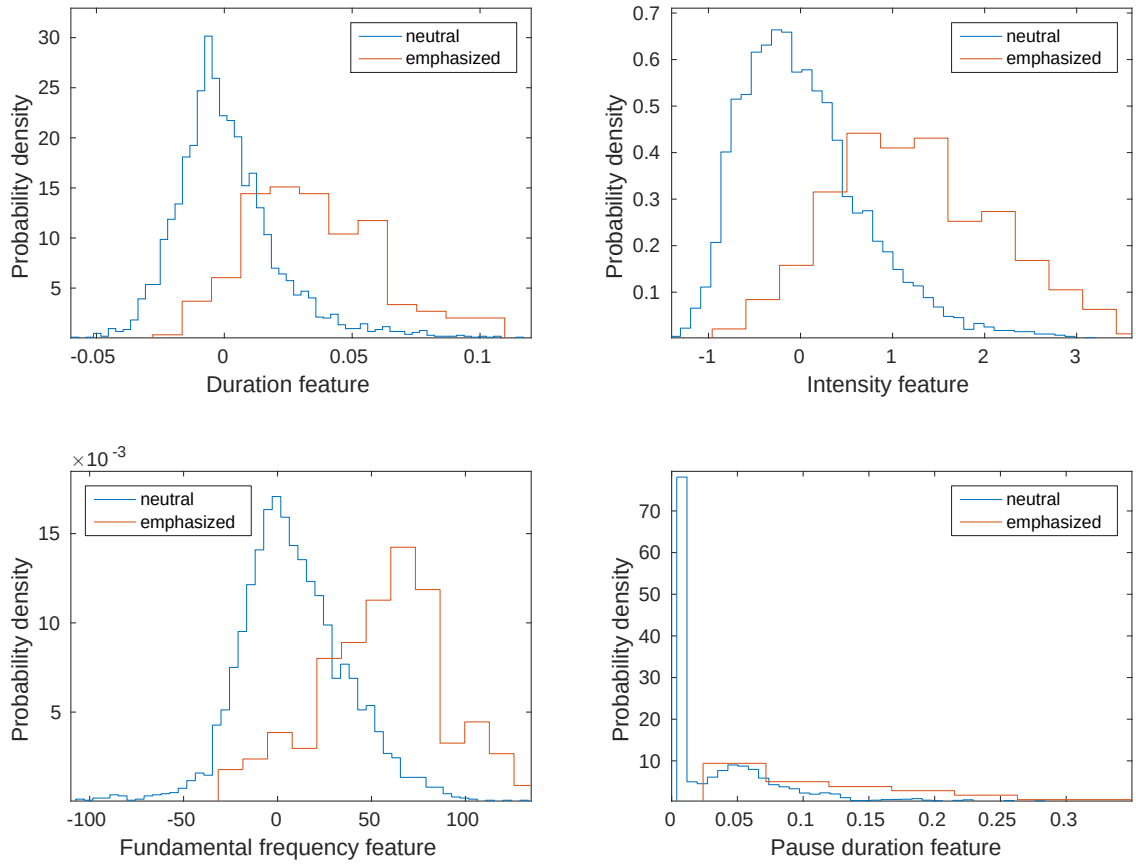



**Figure 3.3:** *Histograms showing the estimated probability distributions of the four derived features of neutral and emphasized words for German*

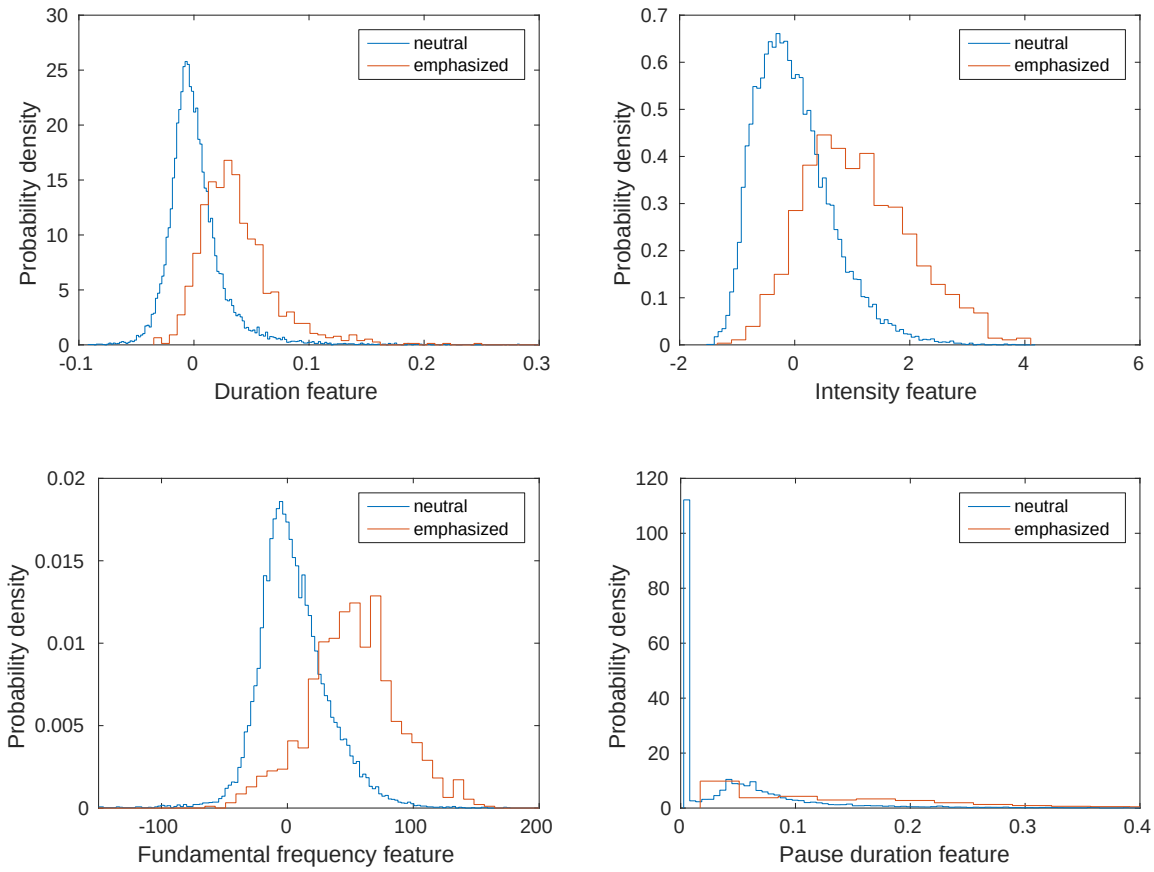



**Figure 3.4:** *Histograms showing the estimated probability distributions of the four derived features of neutral and emphasized words for all languages*

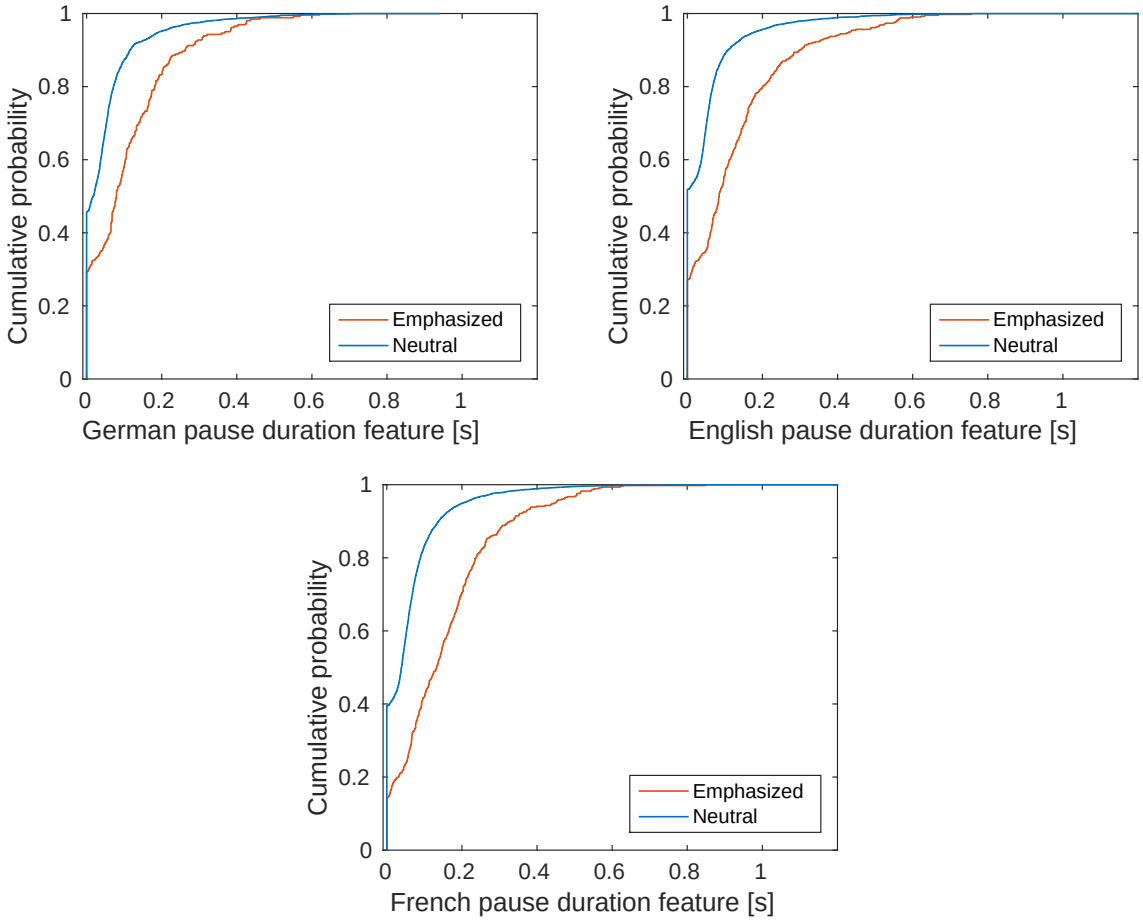



**Figure 3.5:** *Cumulative probability density function of the derived pause duration for neutral and emphasized words*

| | Fisher distance | | |
|---|---|---|---|
| | DE | EN | FR |
| Duration feature | 0.651 | 0.552 | 0.618 |
| $\tau_{dur}$ | 0.049 | 0.057 | 0.037 |
| $\zeta_{dur}$ | 0.012 | 0.124 | 0.129 |

**Table 3.5:** *Fisher distances of the derived duration features*

| | Fisher distance | | |
|---|---|---|---|
| | DE | EN | FR |
| Intensity feature | 0.935 | 1.107 | 0.523 |
| $\tau_{dur}$ | 0.089 | 0.030 | 0.058 |
| $\zeta_{dur}$ | 0.076 | 0.178 | 0.111 |

**Table 3.6:** *Fisher distances of the derived intensity features*

| | Fisher distance | | |
|---|---|---|---|
| | DE | EN | FR |
| Fundamental frequency feature (syllable max) | 0.896 | 0.799 | 0.779 |
| Fundamental frequency feature (syllable average) | 0.538 | 0.419 | 0.645 |

**Table 3.7:** *Fisher distances of the derived fundamental frequency features*

| Properties | | Fisher distance | | |
|---|---|---|---|---|
| Closure adjustment | Positions | DE | EN | FR |
| none | before & after | 0.168 | 0.230 | 0.360 |
| none | after | 0.166 | 0.215 | 0.255 |
| phoneme | before & after | 0.148 | 0.243 | 0.340 |
| phoneme | after | 0.135 | 0.214 | 0.242 |
| plosive class | before & after | 0.145 | 0.246 | 0.341 |
| plosive class | after | 0.132 | 0.216 | 0.241 |

**Table 3.8:** *Fisher distances of the derived pause duration features*

### 3.1.3 Discussion

The histograms and Fisher distances of raw features in Section 3.1.2 show that the extracted raw features do contain information regarding emphasis. Tables 3.1, 3.2 and 3.3 indicate that a phoneme based feature for duration and intensity is not advisable.

The calculation of derived features has further improved the Fisher distances, while drastically reducing data size. The number of features is now four per word as opposed to multiple values per phoneme (one duration value, one intensity value and an array of fundamental frequency values). The derived feature histograms in Figure 3.4 show a clear distinction between the features of emphasized and neutral words. This distinction is even better visible in Figure 3.3, where only one language was considered.

The worst performing derived feature, according to the evaluation methods used in this work, is the pause duration feature. One possible reason for this may be the infrequent occurrence of pauses. Nevertheless, the derived pause feature was still included in the classification step, under the assumption that a classifier would learn to ignore it if the contained information was contradictory. If, in the future, this work's findings are to be implemented as efficiently as possible, the usefulness of including pause duration as a feature should be reassessed.

The Fisher distances presented in Table 3.1 also show, that emphasis does not have the same effect in all three languages. In English for example, raw duration and intensity are much better indicators of emphasis than raw fundamental frequency. For the other languages they are only slightly better. Such differences are to be explained on one hand by the different nature of these languages and on the other hand by the fact that many of the speakers contributing to the English samples of the multilingual SIWIS corpus speak English as a second language.

The language dependency of emphasis manifestation is the reason why in this work classification was only done for each language separately. While a classifier trained with and used for all languages may work, its performance would most likely be considerably lower than that of a language specific classifier.

## 3.2 Classification

### 3.2.1 Evaluation Methods

There exist many types of scores and visualizations used to assess and compare classifier performance. Below, an overview of the ones most relevant to this work is given. It is indicated which ones were used for this work.

**Confusion Matrix** A confusion matrix, as seen in Figure 3.6, shows the relation of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) in a simple, yet structured form. In this work, true positives correspond to emphasized words that have correctly been labeled as such. False positives are neutral words predicted to be emphasized. False negatives are words where the classifier failed to detect emphasis and true negatives are neutral words that were detected as such. In addition to the TP, TN, FP and FN counts, confusion matrices list the following calculations:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Negative Predictive Value} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{14}$$

$$\text{Fallout} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{15}$$

$$\text{False Discovery Rate} = \frac{\text{FP}}{\text{FP} + \text{TP}} \tag{16}$$

$$\text{Miss Rate} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{17}$$

$$\text{False Omission Rate} = \frac{\text{FN}}{\text{TN} + \text{FN}} \tag{18}$$

Confusion matrices summarize classification performance in a comprehensive structure. They are used in this work to visualize and compare the results of the best performing classifiers of each architecture. However, they are not suited for parameter optimization, as parameter optimization calls for single-valued performance measures. It is unclear which of the matrices' values should be compared at what weight to obtain such a measure.

**Accuracy** Accuracy is a frequently used measure of classification performance. However, for the purpose of this thesis it is not suitable. The data set used in this work contains much more neutral words than emphasized words (cf. Table 2.1). As a result, the accuracy scores can become quite high even though only few (or even none) emphasized words have been recognized as such.

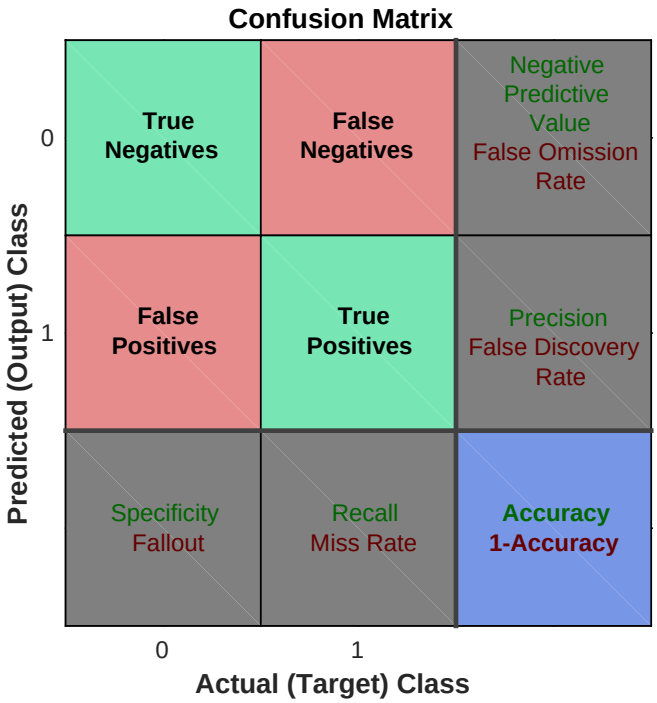



**Figure 3.6:** *Confusion matrix example*

**$F_1$ score** One of the properties, which is most desired in a detection method for emphasized words, is that the classifier detects as many of the emphasized words as possible. This means that a classifier should have high recall. Furthermore, out of the words that are detected as emphasized, as many as possible should truly be emphasized ones. This means that the classifier ought to have high precision.

While both, high precision and recall are desired, there is always a tradeoff between them. As experiments have shown, maximizing one will result in a decrease of the other. Hence, neither of them presented a suitable performance measure for parameter optimization. Instead a combination of the two was used.

A commonly used combination of recall and precision is the $F_1$ score. It is the harmonic mean of precision and recall and calculated as seen in (19). This is the score that was used for parameter optimization.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{19}$$

**ROC curve and AUC** In a receiver operating characeristic (ROC) curve, a classifier's true positive rate for different thresholds is plotted against the corresponding false positive rates. This results in an illustration of how well a binary classifier distinguishes between two classes for different discrimination thresholds.

The area under an ROC curve is called AUC. AUC takes values between zero and one. A low AUC implies bad classification performance, a high one the opposite.

In this work ROC and AUC are used to compare the best classifiers of each of the four architectures.

| | $F_1$ score | | | Accuracy | | |
|---|---|---|---|---|---|---|
| | DE | EN | FR | DE | EN | FR |
| Support vector machines | 0.593 | 0.609 | 0.606 | 0.947 | 0.940 | 0.949 |
| Random forests | 0.607 | 0.613 | 0.577 | 0.946 | 0.942 | 0.946 |
| Hidden Markov models | 0.456 | 0.444 | 0.416 | 0.935 | 0.933 | 0.942 |
| Neural networks | 0.533 | 0.544 | 0.520 | 0.950 | 0.925 | 0.934 |

**Table 3.9:** *Comparison of $F_1$ score and accuracy of the four optimal classifiers*

### 3.2.2 Results

This section summarizes the results of the four different classifiers that were trained for this work.

A comparison of the four optimized classifiers can be found in Table 3.9. In Appendix B some more detailed listings of the five best parameter settings of each individual classifier can be found. The $F_1$ scores, precisions, recalls and accuracies listed below are all mean results from multiple experiments with the same parameter settings. This was done, because some classifiers showed performance deviations between specific instances trained with the same parameter settings. Such deviations are due to differences in initial weights and random division of the samples into folds.

An often observed problem in the training and evaluation of classifiers is overfitting. This is why the classifiers obtained in this work were evaluated both on their training samples and the respective testing samples. Had any of the classifiers been overfitted, this would have shown in the classifier performing much better on the training data than on the testing data. This was not the case for any of the classifiers.

In the individual paragraphs below, one instance of each classifier with the best parameter settings is described more closely. Furthermore an overview of the assessed parameter combinations is given.

#### Support Vector Machines

For support vector machines parameter optimization is very important. Directly applying a MATLAB standard configuraion SVM to the unbalanced data set will result in a support vector machine that classifies all samples to be neutral. This is not desired. To avoid such a one-sided classification, the cost of misclassification was varied. This was achieved by using cost matrices of the form (20), where $\alpha \in [1, 7]$ is the misclassification cost of emphasized samples.

$$\text{cost} = \begin{pmatrix} 0 & 1 \\ \alpha & 0 \end{pmatrix} \tag{20}$$

As an alternative, balanced data sets were used. In the case of balanced data sets a cost matrix of the form (21) with $\alpha \in [1, 7]$ has shown to be beneficial.

$$\text{cost} = \begin{pmatrix} 0 & \alpha \\ 1 & 0 \end{pmatrix} \tag{21}$$

| | Balanced | Standardization | Kernel function | Poly. order | Cost |
|---|---|---|---|---|---|
| DE | no | yes | polynomial | 3 | [0, 1; 3, 0] |
| EN | no | yes | polynomial | 2 | [0, 1; 4, 0] |
| FR | no | yes | polynomial | 3 | [0, 1; 4, 0] |

**Table 3.10:** *Best performing parameter settings for a support vector machine classifier to detect emphasis*

Other parameters that were varied include standardization, kernel function and polynomial order where applicable. Standardization has proven to always be beneficial. The three evaluated kernel functions are linear kernels, radial basis functions and polynomial kernels. As the derived feature vectors are not linearly separable and as the given problem is not a one class problem either, polynomial kernels are expected to perform best. Experiments have confirmed this. Regarding polynomial order, second order polynomials and third order polynomials perform similarly well. However, second order polynomial kernels train significantly faster. Higher order polynomials were not considered due to long calculation times.

An overview of the best support vector machine configurations as found in this work is given in Table 3.10. Figure 3.7 shows the confusion matrices and ROC plots of one instance of a support vector machine with optimal parameters for German, English and French. A more extensive list of the best performing configurations and their average performance scores may be found in Appendix B.

### Random Forests

The optimization of the random forests was done by training many different random forests using varying values for the minimum leaf size, the number of decision split variables and the cost matrix. The minimum leaf size was chosen in the interval $[1, 100]$ with a step size of 10. The number of decision split variables was chosen from the values $1, 2, 3, 4$. The cost matrices were of the form (20) where $\alpha$ varied from 1 to 5.

The results of this optimization can be found in Table 3.11. A list of further well performing parameter settings for random forests in this work can be found in Appendix B. All of the optimal configurations were trained with the original unbalanced training set. The ones trained on a balanced training set, as described in Section 2.4.1, performed considerably worse.

Figure 3.8 shows the confusion matrices and ROC plots of one instance of a random forest which was trained with the optimal parameters listed in Table 3.11.

| | Minimim leaf size | Decision split variables | Cost |
|---|---|---|---|
| DE | 21 | 4 | [0, 1; 3, 0] |
| EN | 11 | 3 | [0, 1; 4, 0] |
| FR | 11 | 3 | [0, 1; 3, 0] |

**Table 3.11:** *Best performing parameter settings for a random forest classifier to detect emphasis*

| | Number of mixture components | Balanced training set |
|---|---|---|
| DE | 1 | true |
| EN | 1 | false |
| FR | 1 | true |

**Table 3.12:** *Best performing parameter settings for a hidden Markov model classifier to detect emphasis*

### Hidden Markov Model

The only parameter to be optimized in the hidden Markov model was the number of mixture components $M$ for the Gaussian mixture model. The optimization was done by running the algorithm with $M$ in the range from one to $24$. Appendix B shows the results of the optimization with the five best parameter settings for the CDHMM.

Table 3.12 summarizes the optimal settings and indicates whether a balanced training set was used. For all languages the CDHMM with only one mixture component performed better than the ones with more mixture components. For German and French the training on a balanced training set achieved optimal results, whereas for English training on an unbalanced training set performed best.

Figure 3.9 shows the confusion matrices and ROC plots for one instance of the CDHMM with the parameters set according to Table 3.12 for German, English and French.

### Neural Networks

For neural networks the most prominent parameters to optimize are the hidden layer dimensions. In Section 2.4.5 the maximum dimensions of a network for the given data set size are calculated. Within the boundaries presented by this calculation the number of neurons was varied at step sizes of one to two for small dimensions and five to ten neurons for higher ones.

An overview of the best network configurations as found in this work is given in Table 3.13. Figure 3.10 shows the confusion matrices and ROC plots of one instance of neural networks

| | Balanced | Hidden layer dimensions |
|---|---|---|
| DE | no | 30 |
| EN | yes | 5-5 |
| FR | yes | 4-4 |

**Table 3.13:** *Best performing parameter settings for a neural network classifier to detect emphasis*

with the optimal parameters for German, English and French. For French, for example, two hidden layers with four neurons each have produced the best average $F_1$ score. A more extensive list of the best performing configurations and their average performance scores may be found in Appendix B.

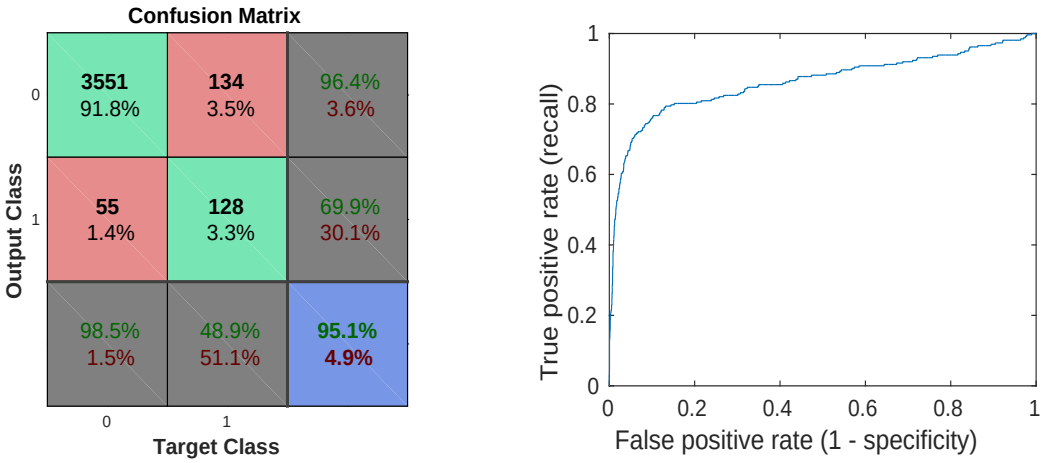
### 3.2.3 Discussion

The results in Section 3.2.2 show that the classification of emphasis is possible with the chosen classifiers.

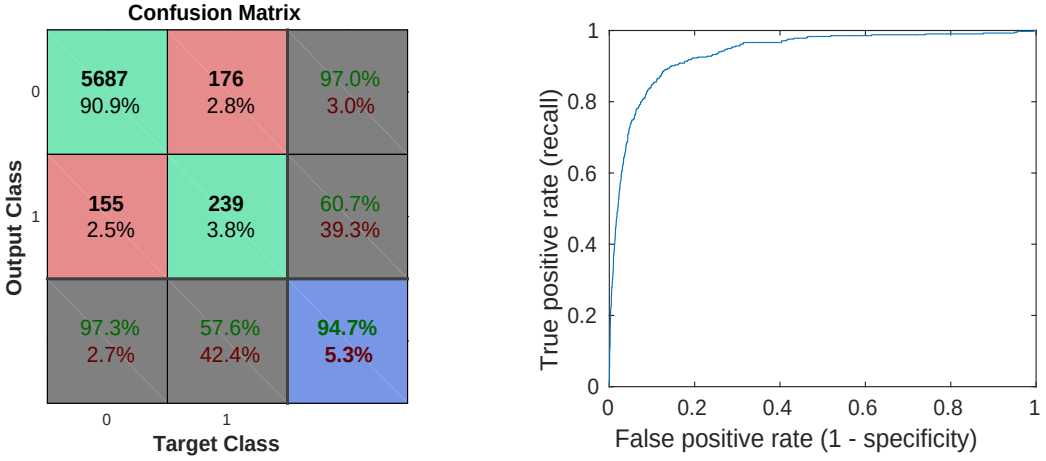
Table 3.9 shows that the best classifier for detecting emphasized words depends on the language. For German and English, random forests achieved the highest $F_1$ score. Support vector machines yielded the best $F_1$ score for French.

The worst performing classifier for all languages is the hidden Markov model. A possible explanation for the mediocre performance of HMMs may be found in the nature of the classifier. An HMM's classification depends on the observation sequence. Its temporal nature tries to learn emphasis sequence patterns. However, the derived features have been designed to eradicate temporal dependencies.

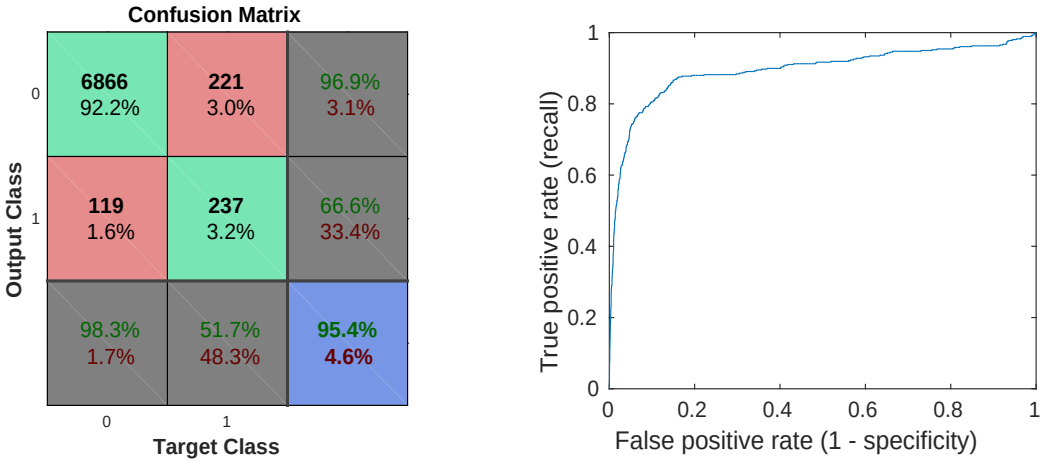
Table 3.9 illustrates, that accuracy is indeed not a favorable performance measure for the classification of emphasized words. With regard to accuracy all classifiers perform well. However, this is also true for a degenerate classifier which classifies all samples to be neutral.




**(a)** German ($AUC = 0.863$)

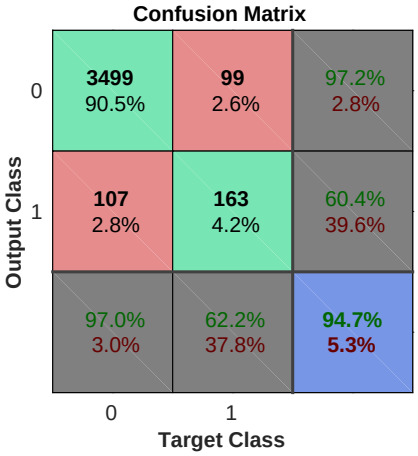


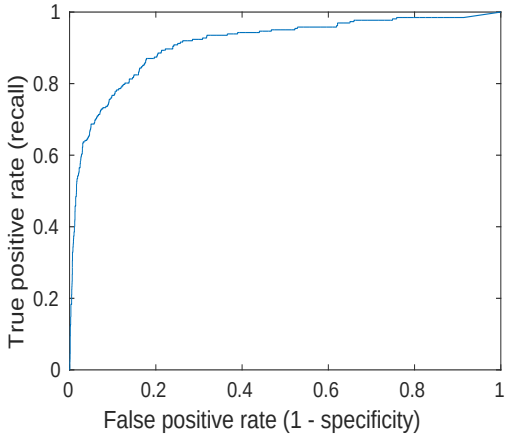

**(b)** English ($AUC = 0.938$)



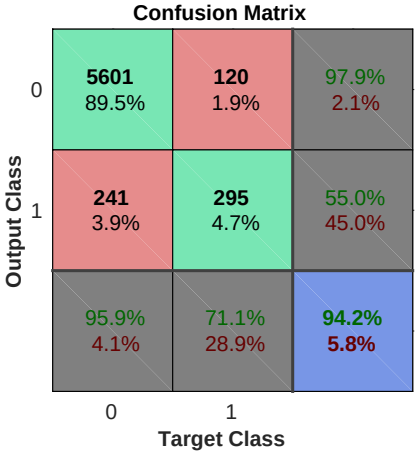

**(c)** French ($AUC = 0.895$)

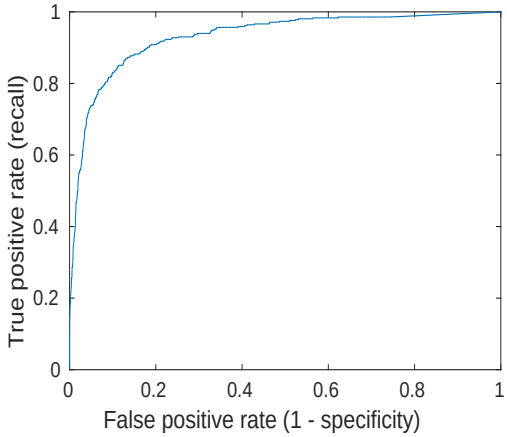
**Figure 3.7:** *Confusion matrices and ROC plots for the optimal support vector machines for German, English and French*



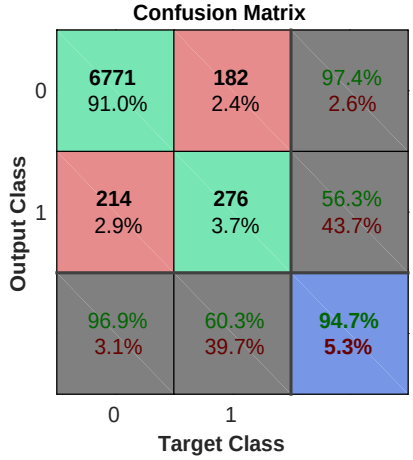



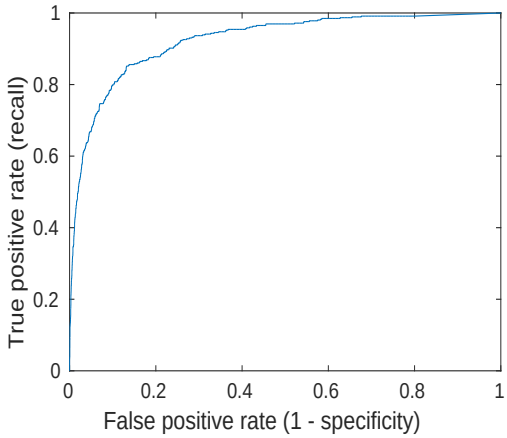

**(a)** German ($AUC = 0.9115$)



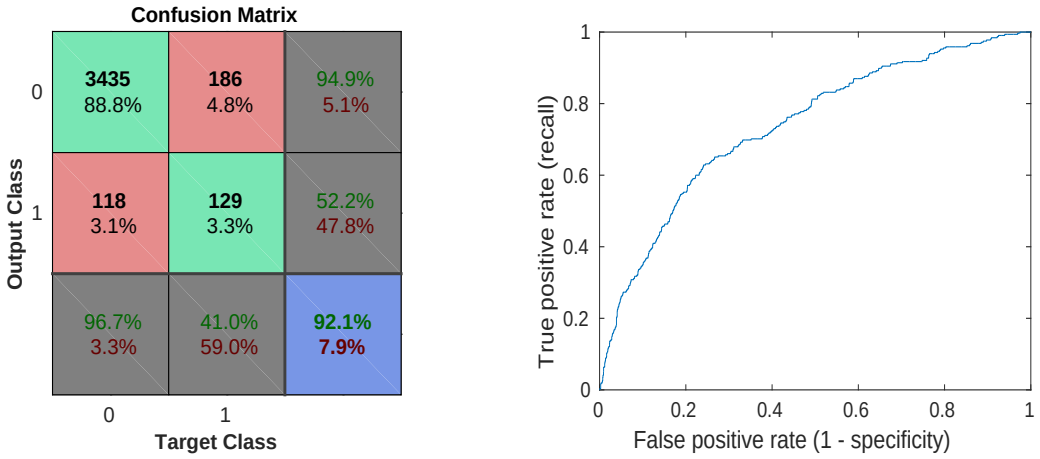

**(b)** English ($AUC = 0.932$)



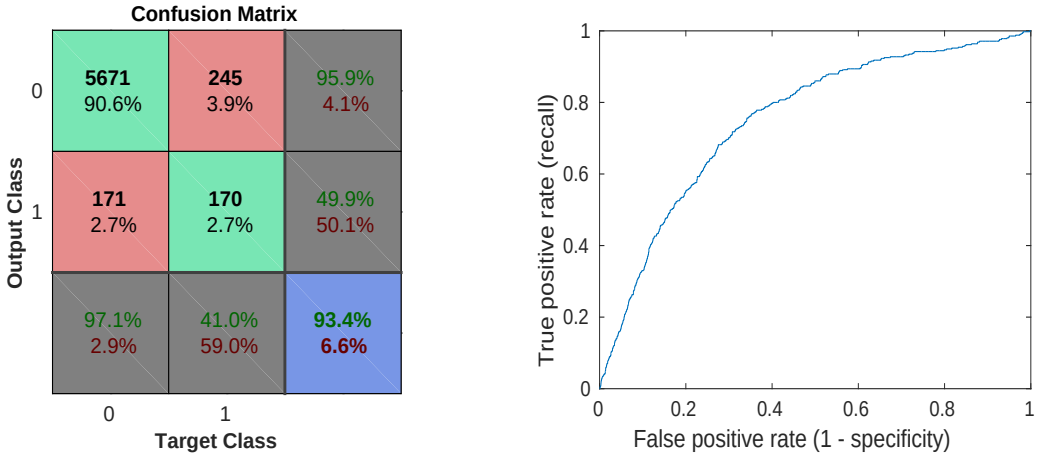




**(c)** French ($AUC = 0.924$)

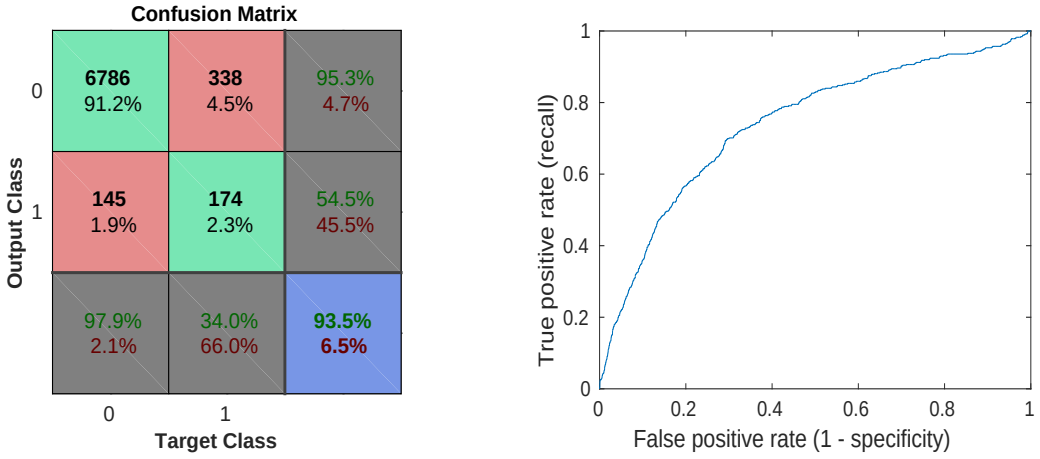
**Figure 3.8:** *Confusion matrices and ROC plots for the optimal random forests for German, English and French*




**(a)** German ($AUC = 0.738$)

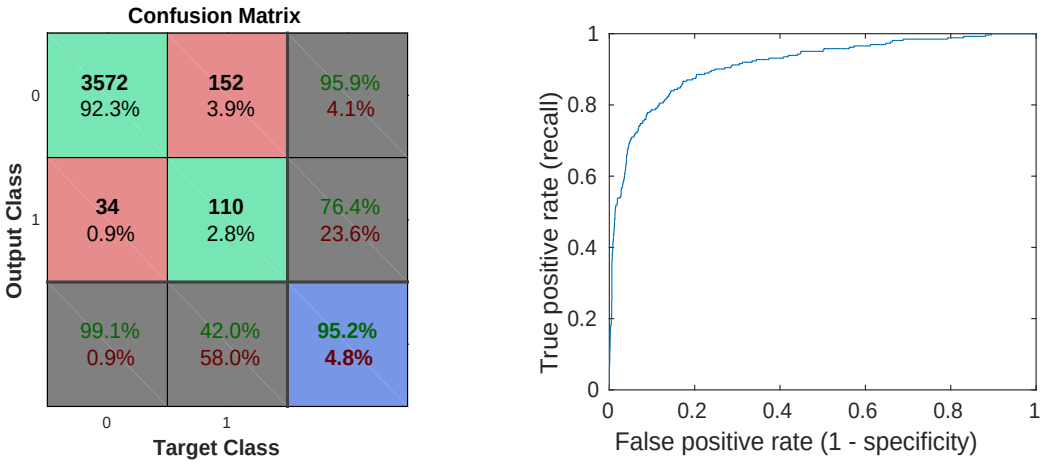



**(b)** English ($AUC = 0.753$)

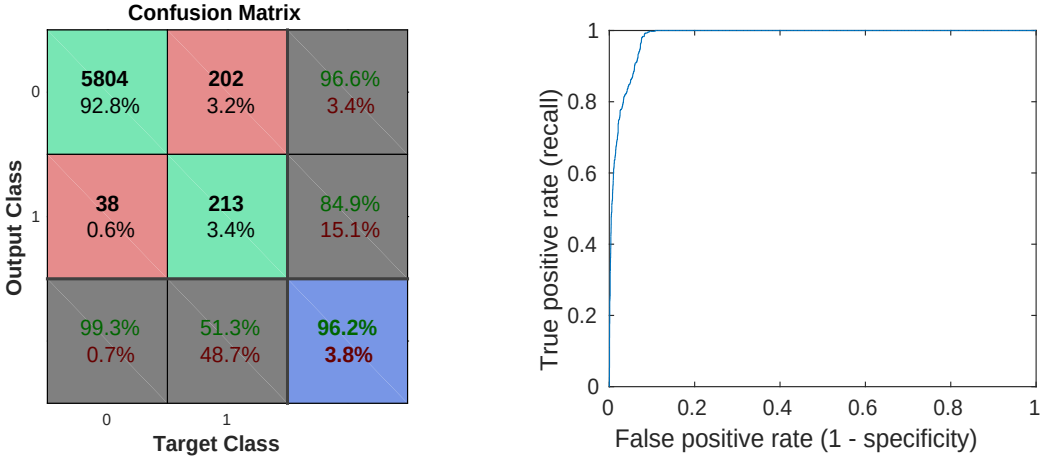



**(c)** French ($AUC = 0.740$)

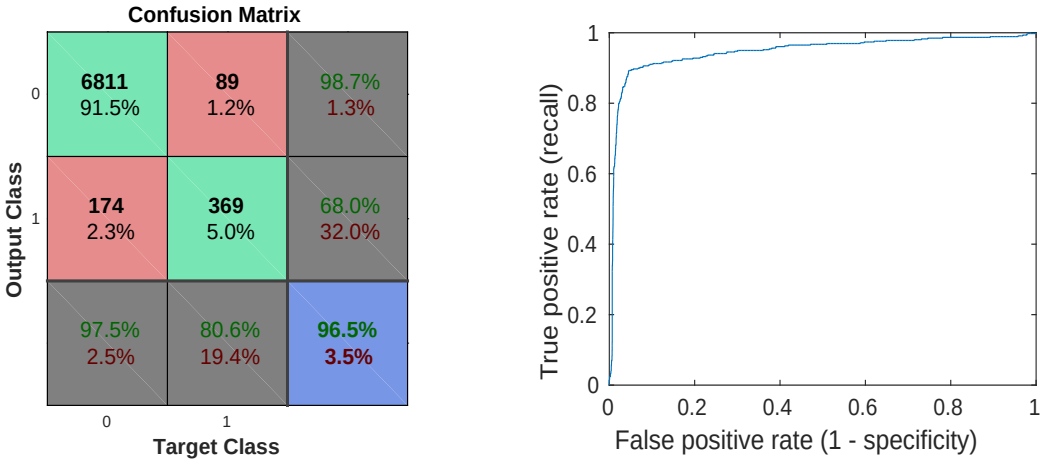
**Figure 3.9:** *Confusion matrices and ROC plots for the optimal HMMs for German, English and French*




**(a)** German ($AUC = 0.893$)




**(b)** English ($AUC = 0.962$)




**(c)** French ($AUC = 0.947$)

**Figure 3.10:** *Confusion matrices and ROC plots for the optimal neural networks for German, English and French*

# 4 Conclusion

This work shows that automatic detection of emphasis is possible.

Raw feature evaluation has shown that the four measurable prosodic quantities (intensity, fundamental frequency, phoneme and pause duration) do contain information on emphasis. For German and English intensity is most strongly influenced by emphasis. Duration is the most prominent carrier of emphasis in French. This shows that emphasis manifestation is language dependent.

Compared to the extracted raw features the derived features found in this work show much higher Fisher distances and more clearly separable probability distributions. For German and English the derived intensity feature results in the best Fisher distances. For French the derived fundamental frequency feature performs best. The worst performing derived feature is pause duration. The inferiority of this feature with comparison to the others may be explained with the infrequent occurrence of pauses.

Emphasis detection classifiers can be trained either for each language individually or for all languages together. Both, for raw features as well as for derived features, the feature with most significant manifestation of emphasis is language dependent. This is why it is to be expected that classifiers will work a lot more efficiently if implemented for each language separately. The most likely application of emphasis detection is to improve prosody in speech-to-speech translation. For translation, the input language needs to be known or detected. Hence, considering that language specific classification ought to outperform a multilingual classifier and that the language to be classified is known, the classification step of this work has been done exclusively for each language separately.

In this work four different types of classifier architecture were assessed. For each of the architectures the respective relevant parameters were optimized. $F_1$ scores were chosen as optimization variables. This score takes into account both recall and precision at equal rates.

With regard to $F_1$ scores, random forests have proven to be the best classification method for German and English. Support vector machines are the best classification method for French. As the scores for both classification methods are very similar for all languages, they may be used interchangeably. A decision between one and the other may be based on memory and time requirements. Out of hidden Markov models and neural networks, neural networks perform better.

If the findings in this work are to be used in the future, the chosen classification method and parameters should be matched to the task at hand. In this work, parameters have been optimized using $F_1$ scores. This results in equal weight of precision and recall. For some tasks, such as speech-to-speech translation, high precision may be of greater importance than recall, since a wrongly emphasized word may lead to more misunderstandings than omitting an emphasis. However, in a case where emphasis is the key feature, recall may be more important.

# 5 Future Work

There are several possibilities to extend or improve the findings in this work. Below, a few of them are summarized shortly.

- The emphasis of this work is on classification performance, not efficiency. If the emphasis detection of this work is to be utilized in a system where time and memory considerations are of importance, the feature extraction and classification methods should be evaluated and adapted accordingly.
- Classifier parameter optimization is calculation intensive. Hence, only a finite set of parameter combinations were evaluated in this work. It is possible that trying further combinations may lead to better results.
- Feature extraction and classification were done separately in this work. Feature extraction was optimized using Fisher distances and then suitable classifiers were found. This was to avoid an optimization with too many parameters (both from the classifiers and the feature extraction). Now that a set of well functioning classifiers is given, it would be possible to reexamine the choice of features. Modified features could now be directly assessed by their classification results, using the gained knowledge on suitable classifier parameters.

In the end, the authors would like to thank the SIWIS project (SNSF Grant CRSII2-141903) for providing multilingual speech data for all the experiments carried out in this thesis.

# A Feature Histograms

This chapter presents the histograms of the raw and derived features of the languages which were not shown in Section 3.1.

## A.1 Raw Features

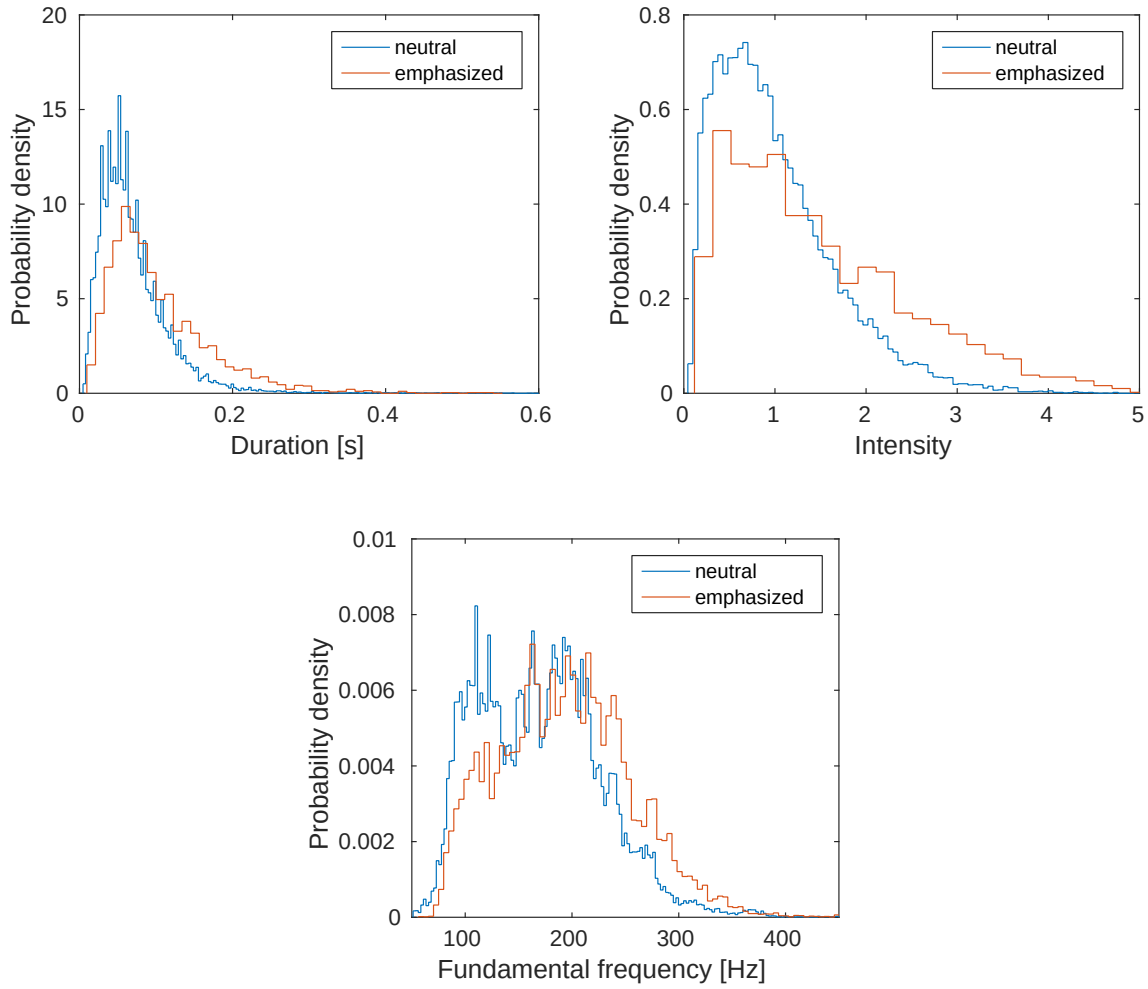



**Figure A.1:** *Histograms showing the estimated probability distributions of the three raw features of neutral and emphasized words for English*

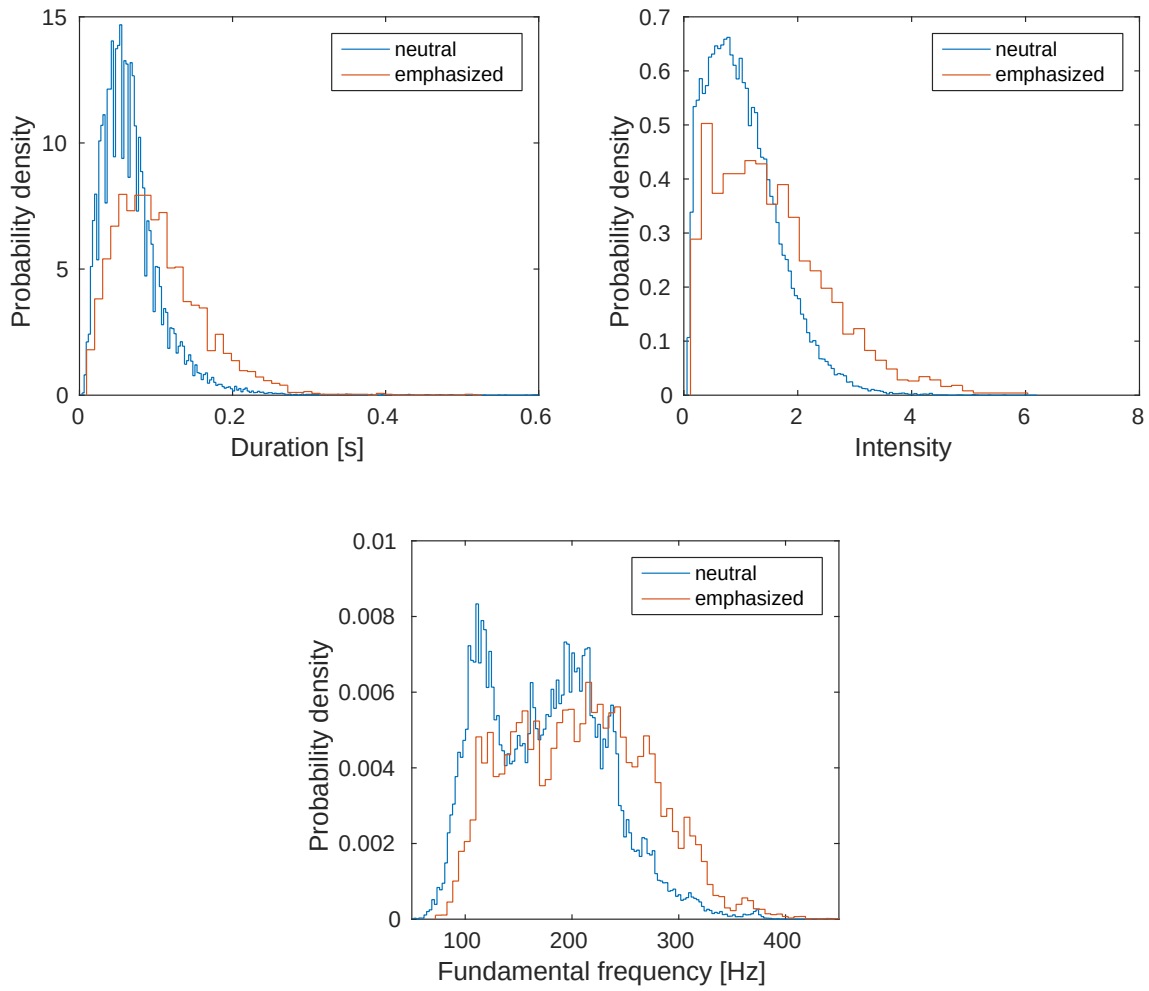



**Figure A.2:** *Histograms showing the estimated probability distributions of the three raw features of neutral and emphasized words for French*

## A.2 Derived Features

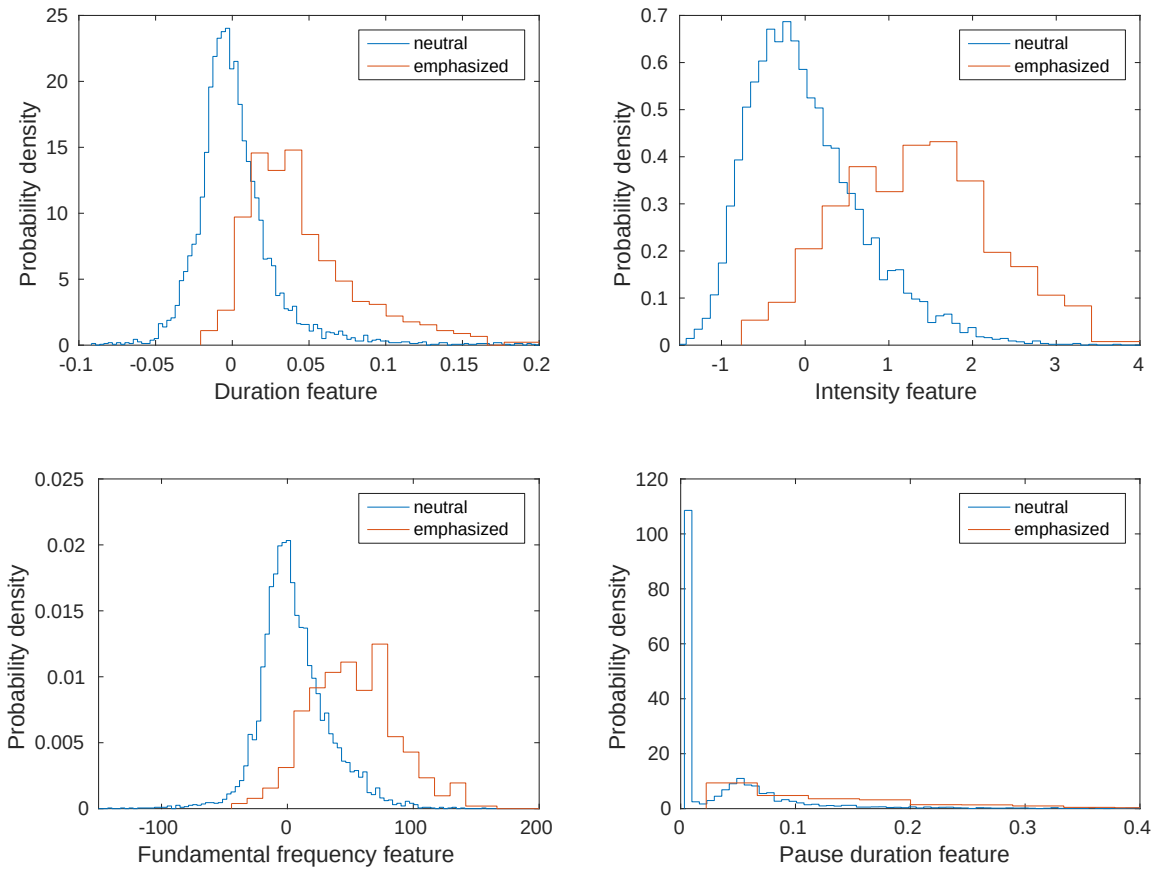



**Figure A.3:** *Histograms showing the estimated probability distributions of the four derived features of neutral and emphasized words for English*

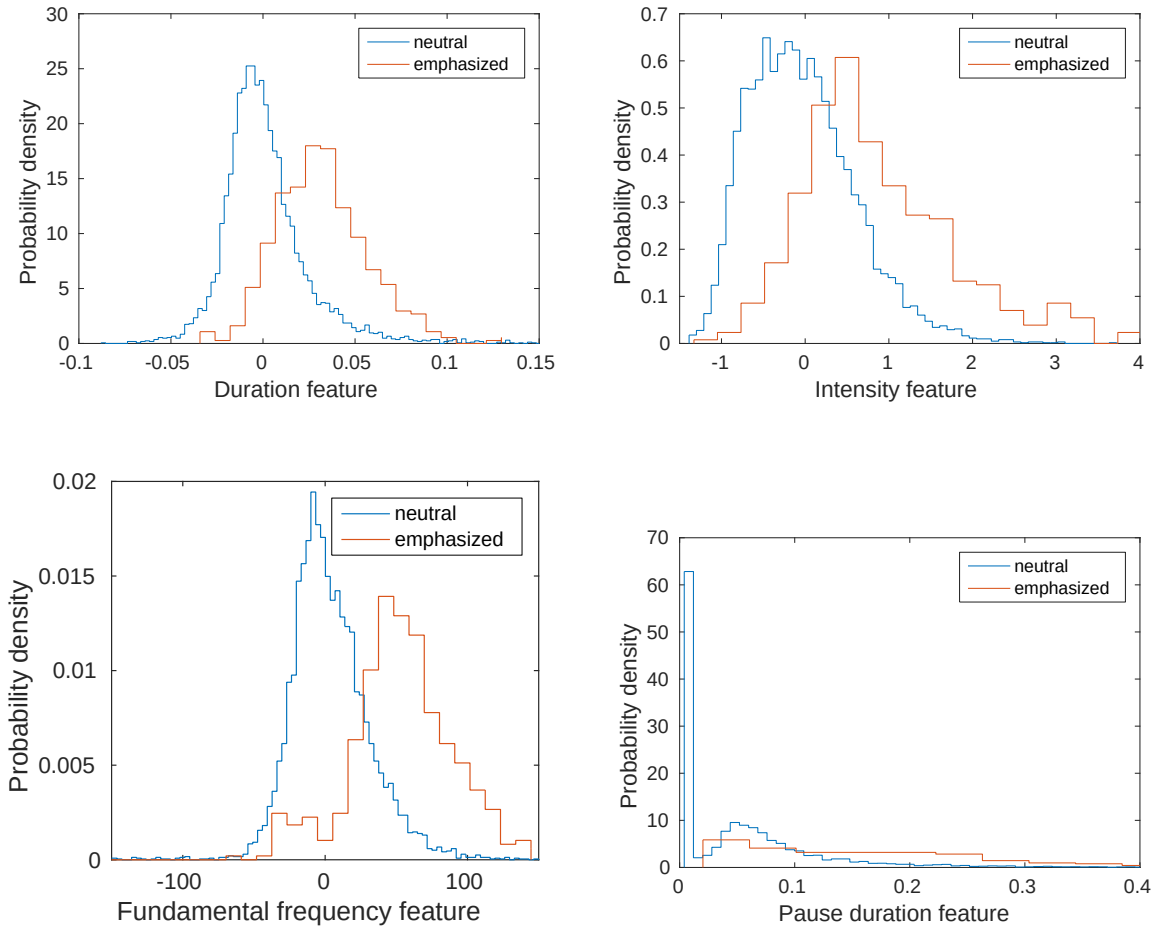



**Figure A.4:** *Histograms showing the estimated probability distributions of the four derived features of neutral and emphasized words for French*

# B Classifier Optimization

This chapter shows the results of the optimization of the four different classifiers. The numbers presented in the following tables are mean values over several experiments with the same configuration. Thus, calculating the $F_1$ score from the listed precision and recall might not always give the same number as the listed $F_1$ score.

## B.1 Support Vector Machines

Note that a support vector machine classifer using the MATLAB standard configuration will predict all samples to be neutral. Hence, precision and $F_1$ score are not defined.

| | $F_1$ | Precision | Recall | Bal. | Standard-ization | Kernel function | Cost | Poly. order |
|---|---|---|---|---|---|---|---|---|
| Configuration 1 | 0.593 | 0.613 | 0.574 | no | yes | poly. | [0, 1;3, 0] | 3 |
| Configuration 2 | 0.593 | 0.682 | 0.524 | no | yes | poly. | [0, 1;2, 0] | 2 |
| Configuration 3 | 0.592 | 0.562 | 0.626 | no | yes | poly. | [0, 1;4, 0] | 3 |
| Configuration 4 | 0.591 | 0.561 | 0.625 | no | yes | poly. | [0, 1;4, 0] | 2 |
| Configuration 5 | 0.587 | 0.563 | 0.613 | yes | yes | poly. | [0, 6;1, 0] | 2 |
| Standard config. | n.a. | n.a. | 0 | no | no | linear | [0, 1; 1, 0] | n.a. |

**Table B.1:** *5 best parameter settings for support vector machines for German compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Bal. | Standard-ization | Kernel function | Cost | Poly. order |
|---|---|---|---|---|---|---|---|---|
| Configuration 1 | 0.609 | 0.539 | 0.700 | no | yes | poly. | [0, 1;4, 0] | 2 |
| Configuration 2 | 0.605 | 0.574 | 0.640 | no | yes | poly. | [0, 1;3, 0] | 2 |
| Configuration 3 | 0.601 | 0.590 | 0.615 | yes | yes | poly. | [0, 6;1, 0] | 3 |
| Configuration 4 | 0.601 | 0.529 | 0.695 | no | yes | poly. | [0, 1;4, 0] | 3 |
| Configuration 5 | 0.599 | 0.502 | 0.742 | no | yes | poly. | [0, 1;5, 0] | 2 |
| Standard config. | n.a. | n.a. | 0 | no | no | linear | [0, 1; 1, 0] | n.a. |

**Table B.2:** *5 best parameter settings for support vector machines for English compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Bal. | Standard-ization | Kernel function | Cost | Poly. order |
|---|---|---|---|---|---|---|---|---|
| Configuration 1 | 0.606 | 0.539 | 0.638 | no | yes | poly. | [0, 1;4, 0] | 3 |
| Configuration 2 | 0.603 | 0.590 | 0.592 | no | yes | poly. | [0, 1;3, 0] | 2 |
| Configuration 3 | 0.599 | 0.555 | 0.649 | no | yes | poly. | [0, 1;4, 0] | 2 |
| Configuration 4 | 0.597 | 0.624 | 0.572 | no | yes | poly. | [0, 1;3, 0] | 3 |
| Configuration 5 | 0.593 | 0.526 | 0.679 | no | yes | poly. | [0, 1;5, 0] | 3 |
| Standard config. | n.a. | n.a. | 0 | no | no | linear | [0, 1; 1, 0] | n.a. |

**Table B.3:** *5 best parameter settings for support vector machines for French compared to the standard MATLAB configuration*

## B.2 Random Forests

| | $F_1$ | Precision | Recall | Minimim leaf size | Decision split variables | Cost |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.607 | 0.595 | 0.619 | 21 | 4 | [0, 1; 3, 0] |
| Configuration 2 | 0.599 | 0.645 | 0.559 | 21 | 4 | [0, 1; 2, 0] |
| Configuration 3 | 0.596 | 0.590 | 0.603 | 11 | 3 | [0, 1; 3, 0] |
| Configuration 4 | 0.596 | 0.564 | 0.632 | 11 | 3 | [0, 1; 4, 0] |
| Configuration 5 | 0.595 | 0.550 | 0.649 | 21 | 4 | [0, 1; 4, 0] |
| Standard config. | 0.551 | 0.706 | 0.452 | 1 | 2 | [0, 1; 1, 0] |

**Table B.4:** *5 best parameter settings for random forests for German compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Minimim leaf size | Decision split variables | Cost |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.613 | 0.547 | 0.699 | 11 | 3 | [0, 1; 4, 0] |
| Configuration 2 | 0.610 | 0.530 | 0.718 | 21 | 4 | [0, 1; 4, 0] |
| Configuration 3 | 0.610 | 0.554 | 0.677 | 21 | 4 | [0, 1; 3, 0] |
| Configuration 4 | 0.609 | 0.530 | 0.715 | 11 | 3 | [0, 1; 5, 0] |
| Configuration 5 | 0.607 | 0.560 | 0.662 | 11 | 3 | [0, 1; 3, 0] |
| Standard config. | 0.560 | 0.667 | 0.483 | 1 | 2 | [0, 1; 1, 0] |

**Table B.5:** *5 best parameter settings for random forests for English compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Minimim leaf size | Decision split variables | Cost |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.577 | 0.558 | 0.597 | 11 | 3 | [0, 1; 3, 0] |
| Configuration 2 | 0.571 | 0.603 | 0.543 | 11 | 3 | [0, 1; 2, 0] |
| Configuration 3 | 0.568 | 0.523 | 0.623 | 11 | 3 | [0, 1; 4, 0] |
| Configuration 4 | 0.560 | 0.499 | 0.637 | 11 | 3 | [0, 1; 5, 0] |
| Configuration 5 | 0.547 | 0.565 | 0.531 | 1 | 2 | [0, 1; 4, 0] |
| Standard config. | 0.536 | 0.668 | 0.448 | 1 | 2 | [0, 1; 1, 0] |

**Table B.6:** *5 best parameter settings for random forests for French compared to the standard MATLAB configuration*

## B.3 Hidden Markov Models

Note that the presented hidden Markov model was self-implemented and therefore does not have a standard MATLAB configuration.

| | $F_1$ | Precision | Recall | Number of mixture components | Balanced training set |
|---|---|---|---|---|---|
| Configuration 1 | 0.456 | 0.525 | 0.403 | 1 | yes |
| Configuration 2 | 0.423 | 0.501 | 0.366 | 1 | no |
| Configuration 3 | 0.351 | 0.760 | 0.229 | 3 | yes |
| Configuration 4 | 0.334 | 0.775 | 0.213 | 4 | yes |
| Configuration 5 | 0.331 | 0.739 | 0.213 | 9 | yes |

**Table B.7:** *5 best parameter settings for HMM for German*

| | $F_1$ | Precision | Recall | Number of mixture components | Balanced training set |
|---|---|---|---|---|---|
| Configuration 1 | 0.444 | 0.492 | 0.405 | 1 | no |
| Configuration 2 | 0.306 | 0.522 | 0.217 | 2 | yes |
| Configuration 3 | 0.302 | 0.639 | 0.198 | 3 | no |
| Configuration 4 | 0.292 | 0.628 | 0.190 | 5 | yes |
| Configuration 5 | 0.284 | 0.622 | 0.184 | 13 | no |

**Table B.8:** *5 best parameter settings for HMM for English*

| | $F_1$ | Precision | Recall | Number of mixture components | Balanced training set |
|---|---|---|---|---|---|
| Configuration 1 | 0.416 | 0.541 | 0.338 | 1 | yes |
| Configuration 2 | 0.349 | 0.448 | 0.285 | 1 | no |
| Configuration 3 | 0.346 | 0.624 | 0.239 | 3 | yes |
| Configuration 4 | 0.343 | 0.667 | 0.231 | 4 | yes |
| Configuration 5 | 0.341 | 0.684 | 0.227 | 6 | yes |

**Table B.9:** *5 best parameter settings for HMM for French*

## B.4 Neural Networks

| | $F_1$ | Precision | Recall | Normalized | Balanced training set | Hidden layer dimensions |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.533 | 0.728 | 0.421 | no | no | 30 |
| Configuration 2 | 0.519 | 0.713 | 0.410 | no | no | 65 |
| Configuration 3 | 0.516 | 0.722 | 0.402 | no | no | 60 |
| Configuration 4 | 0.513 | 0.713 | 0.402 | no | no | 40 |
| Configuration 5 | 0.512 | 0.735 | 0.394 | no | no | 50 |
| Standard config. | 0.423 | 0.610 | 0.354 | no | no | 10 |

**Table B.10:** *5 best parameter settings for neural networks for German compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Normalized | Balanced training set | Hidden layer dimensions |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.544 | 0.460 | 0.773 | no | yes | 5-5 |
| Configuration 2 | 0.505 | 0.684 | 0.401 | no | no | 50 |
| Configuration 3 | 0.495 | 0.697 | 0.385 | no | no | 65 |
| Configuration 4 | 0.489 | 0.671 | 0.386 | no | no | 60 |
| Configuration 5 | 0.489 | 0.373 | 0.807 | yes | yes | 2 |
| Standard config. | 0.423 | 0.632 | 0.324 | no | no | 10 |

**Table B.11:** *5 best parameter settings for neural networks for English compared to the standard MATLAB configuration*

| | $F_1$ | Precision | Recall | Normalized | Balanced training set | Hidden layer dimensions |
|---|---|---|---|---|---|---|
| Configuration 1 | 0.520 | 0.477 | 0.710 | no | yes | 4-4 |
| Configuration 2 | 0.510 | 0.721 | 0.396 | no | no | 60 |
| Configuration 3 | 0.507 | 0.716 | 0.716 | no | no | 55 |
| Configuration 4 | 0.505 | 0.720 | 0.390 | no | no | 50 |
| Configuration 5 | 0.503 | 0.700 | 0.393 | no | no | 20-20 |
| Standard config. | 0.465 | 0.719 | 0.345 | no | no | 10 |

**Table B.12:** *5 best parameter settings for neural networks for FR compared to the standard MATLAB configuration*

# C Phone Inventories

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

## Definition of the Phone Inventories to be Used for Mixed ingual TTS Synthesis

Beat Pfister

June 1, 2010

The following pages define the phone inventories for German, French, Italian, English, and Spanish as used e.g. in the mixed-lingual SVO TTS system. Th e phones are represented by means of IPA symbols and by symbols of the ETH computer phonetic alphabet (ETHPA). Each phone or diphtong is illustrated with some examples in graphemic and phonetic form.

For readability reasons, the ETHPA symbols are defined to be as similar as possible to the IPA symbols. IPA symbols as well as ETHPA symbols representing the phonetic transcription of a word can be put in a strings. Such word strings can unambiguously be split into phonemes (not true for most other phonetic alpabet such as SAMPA).

The phonetic forms of words in IPA symbols may be produced in LaTeX documents by means of the command `string`, where `string` is a sequence of ETHPA symbols from the lists below. Note that the string must have no blanks; they will not appear in the phonetic form on the output document.

### Hints for lexicon writers

- The following phonetic dictionaries are considered as standard for lexical entries: for German [Dud74]; for French [War87]; for Italian [Pon95]; for American and British English [Jon03]; and for Spanish [Lan02].
- Apart from French, every stem and fullform entry must get a main word accent, denoted as single quote (i.e. )
- Dashes or hyphens of graphemic representations are to be replaced by the symbol in the phonetic representations (e.g. the pronunciation of "Punt-Chamues-ch" to appear in the lexicon is: `pntt_amust_`)
- Proper names consisting of several words (e.g. "Segl Baselgia") go in one single lexicon entry. The pronunciations of the words are also separated blanks, i.e. `se bas:ld_a` .
- Aspiration of German and English plosives (i.e. [$k^h$], [$p^h$] and [$t^h$]) is not indicated in the lexicon; it can be determined by means of appropriate rules.

- The German syllabic consonants (i.e. [l̩], [m̩] and [n̩]) must not be used in the lexicon; use the pronunciation with the Schwa instead.

## References

[Dud74] *Duden "Aussprachew rterbuch",* 2. Auflage. Bibliographisches Institut. Mannheim, Wien, Z rich, 1974.

[Jon03] D. Jones. *Cambridge English Pronouncing Dictionary*. Cambridge University Press (ISBN 0-521-01712-2), 16th edition, 2003.

[Lan02] *Langenscheidt "Maxi-W rterbuch Spanisch"* . Langenscheidt Verlag Berlin und M nchen (ISBN 3-468-11441-9), 2002.

[Pon95] *Pons- ompaktw rterbuch Italienisch-Deutsch,* 2. Auflage. Ernst Klett Verlag, 1995.

[War87] L. Warnant. *Dictionnaire de la prononciation fran aise* . Duculot (ISBN 2-8011-0581-3), 1987.

## erman phone inventory (incl. Swiss German diphthongs)

| IPA | ETHPA | Example | |
|---|---|---|---|
| aː | a: | Bahn | [ˈbaːn] |
| a | a | hat | [ˈhat] |
| ɐ | | Ober | [ˈʔoːbɐ] |
| ɐ̯ | | Uhr | [ˈʔuːɐ̯] |
| a͡i | a_i | weit | [ˈva͡it] |
| a͡u | a_u | Haut | [ˈha͡ut] |
| b | b | Ball | [ˈbal] |
| | | ich | [ˈʔɪ ] |
| d | d | dann | [ˈdan] |
| d͡ʒ | d_ | Gin | [ˈd͡ʒɪn] |
| eː | e: | Beet | [ˈbeːt] |
| e | e | Methan | [meˈtaːn] |
| ɛː | : | w hle | [ ˈvɛːlə] |
| ɛ | | h tte | [ ˈhɛtə] |
| e͡ɪ | e_ | Frey | [ˈfre͡ɪ] [2] |
| ə | | halte | [ˈhaltə] |
| f | f | Fass | [ˈfas] |
| g | g | Gast | [ˈgast] |
| g͡g | g_g | R egger | [ ˈryə͡ggər] [3] |
| h | h | hat | [ˈhat] |
| iː | i: | viel | [ˈfiːl] |
| i | i | vital | [viˈtaːl] |
| ɪ | | bist | [ˈbɪst] |
| i̯ | i | Studie | [ˈʃtuːdi̯e] |
| i̯ə | i_ | Dietikon | [ˈdi̯əti̯ˌkoːn] [2] |
| j | j | ja | [ˈjaː] |
| k | k | Skandal | [skanˈdaːl] |
| kʰ | k_h | kalt | [ˈkʰalt] [1] |
| l | l | Last | [ˈlast] |
| l̩ | l | Nabel | [ˈnaːbl̩] |
| m | m | Mast | [ˈmast] |
| m̩ | m | grossem | [ˈgroːsm̩] |
| n | n | Naht | [ˈnaːt] |
| n̩ | n | baden | [ˈbaːdn̩] |
| ŋ | | lang | [ˈlaŋ] |
| oː | o: | Boot | [ˈboːt] |
| o | o | Moral | [moˈraːl] |
| ɔ | | Post | [ˈpɔst] |
| o̯ | o | loyal | [lo̯aˈjaːl] |
| ɔ͡y | _y | Heu | [ˈhɔ͡y] |

| IPA | ETHPA | Example | |
|---|---|---|---|
| ː | : | l | [ ˈʔ ːl] |
| | | konom | [ ˌʔ ko ˈnoːm] |
| | | göttlich | [ˈg tl ɪ ] |
| p | p | Spatz | [ˈʃpat͡s] |
| p͡f | p_f | Pfahl | [ˈp͡faːl] |
| pʰ | p_h | Pakt | [ˈpʰakt] [1] |
| r | r | Rast | [ˈrast] |
| r͡r | r_r | Karren | [ˈkar͡rən] |
| s | s | Hast | [ˈhast] |
| ʃ | | Schal | [ˈʃaːl] |
| t | t | Stier | [ˈʃtiːr] |
| tʰ | t_h | Tal | [ˈtʰaːl] [1] |
| t͡s | t_s | Zahl | [ˈt͡saːl] |
| t͡ʃ | t_ | Matsch | [ˈmat͡ʃ] |
| uː | u: | Hut | [ˈhuːt] |
| u | u | kulant | [kuˈlant] |
| ʊ | | Pult | [ˈpʰʊlt] |
| u̯ | u | aktuell | [akˈtu̯ɛl] |
| u̯ə | _ | Ruedi | [ˈru̯ə̯di] [2] |
| v | v | was | [ˈvas] |
| x | x | Bach | [ˈbax] |
| yː | y: | R be | [ ˈryːbə] |
| y | y | Mykene | [myˈkeːnə] |
| ʏ | | f llt | [ ˈfʏlt] |
| y | y | Etui | [ʔeˈtyiː] |
| y̯ə | y_ | Bl emlisalp | [ ˌblyə̯mlisˈalp] [2] |
| z | z | Hase | [ˈhaːzə] |
| ʒ | | Genie | [ʒeˈniː] |
| ʔ | | beamtet | [bəˈʔamtət] |

[1] aspirated plosive (not in lexicon)
[2] Swiss German diphthong
[3] strong Swiss German [g]

## French phone inventory

| IPA | ETHPA | Example | | |
|---|---|---|---|---|
| a | a | tabac | [taba] | [1] optional schwa |
| ɑ | | b t, p te | [b ɑ], [pɑt] | |
| | a | ange | [ ʒ] | |
| b | b | bon, robe | [bɔ], [ʁɔb] | |
| d | d | dans, aide | [d ], [ ɛd] | |
| e | e | t | [ete] | |
| ɛ | | treize | [tʁɛz] | |
| ɛ | | cinq, linge | [zɛk], [lɛʒ] | |
| ə | | premier | [pʁəmje] | |
| (ə) | | matelot | [mat(ə)lo] [1] | |
| f | f | feu, neuf | [f ], [n f] | |
| g | g | gare, bague | [gaʁ], [bag] | |
| h | h | hop | [hɔp] | |
| i | i | lit, mis | [li], [emi] | |
| j | j | yeux, paille | [j ], [paj] | |
| ɲ | | agneau, vigne | [aɲo], [viɲ] | |
| k | k | actif, barque | [aktif], [baʁk] | |
| l | l | lent, sol | [l ], [s ɔl] | |
| m | m | main, femme | [mɛ], [fɑm] | |
| ŋ | | camping | [k pi ŋ] | |
| n | n | nous, tonne | [nu], [tɔn] | |
| o | o | galop | [galo] | |
| ɔ | | loge | [el ɔʒ] | |
| ɔ | | on, savon | [ɔn], [savɔ] | |
| | | bleu | [bl ] | |
| | | neuf, oeuf | [n f], [ f] | |
| | | un, parfum | [ ], [pa ʁf ] | |
| p | p | p re, soupe | [p ɛʁ], [sup] | |
| ʁ | R | rue, venir | [ʁy], [vəniʁ] | |
| s | s | sale, dessous | [sal], [dəsu] | |
| ʃ | | chat, t che | [ ʃa], [taʃ] | |
| t | t | terre, vite | [tɛʁ], [vit] | |
| u | u | roue | [ʁu] | |
| v | v | vous, r ve | [vu], [ ʁɛv] | |
| w | w | oui, nouer | [wi], [nwe] | |
| y | y | lu | [ly] | |
| ɥ | | huit, lui | [ɥit], [lɥi] | |
| z | z | z ro, maison | [ze ʁo], [mɛzɔ] | |
| ʒ | | gilet, mijoter | [ʒilɛ],[miʒɔte] | |
| ʔ | | les haricots | [le ʔaʁiko] | |

## English phone inventory

| IPA | ETHPA | Example | |
|---|---|---|---|
| ə | | another | [əˈnʌ ə] |
| əʊ | _ | nose | [ˈnəʊz] [1] |
| | q | hat | [ˈh t] |
| ɑ | | got, frog | [ˈgɑt], [ˈfrɑg] [2] |
| ɑː | : | stars | [ˈstɑːz] [1], [ˈstɑːrz] [2] |
| ʌ | | cut, much | [ˈkʌt], [ˈmʌtʃ] |
| aɪ | a_ | rise | [ˈraɪz] |
| aʊ | a_ | about | [əˈbaʊt] |
| b | b | bin | [ˈbɪn] |
| | | this, other | [ˈ ɪs], [ˈʌ ər] |
| d | d | din | [ˈdɪn] |
| dʒ | d_ | Gin | [ˈdʒɪn] |
| ɜː | : | bird, furs | [ˈbɜːd], [ˈfɜːz] [1] |
| ɜ | | bird, furs | [ˈbɜrd], [ˈfɜrz] [2] |
| e | e | get | [ˈget] |
| eɪ | e_ | raise | [ˈreɪz] |
| ɛə | _ | stairs | [ˈstɛəz] [1], [ˈstɛərz] [2] |
| f | f | fit | [ˈfɪt] |
| g | g | give, bag | [ˈgɪv], [ˈb g] |
| h | h | hit | [ˈhɪt] |
| ɪ | | witch | [ˈwɪtʃ] |
| iː | i: | ease | [ˈiːz] |
| ɪə | _ | 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 | | thing | [ˈθɪŋ] |
| n | n | fine, net | [ˈfaɪn], [ˈnet] |
| ɔː | : | abroad | [əˈbrɔːd] |
| ɔɪ | _ | noise | [ˈnɔɪz] |
| ɒ | Q | got, frog | [ˈgɒt], [ˈfrɒg] [1] |
| oʊ | o_ | nose | [ˈnoʊz] [2] |
| p | p | speed | [ˈspiːd] |
| pʰ | p_h | pin | [ˈpʰɪn] |
| r | r | ring, stress | [ˈrɪŋ], [ˈstres] |
| ʃ | | shine, brush | [ˈʃaɪn], [ˈbrʌʃ] |
| s | s | sin, mouse | [ˈsɪn], [ˈmaʊs] |
| θ | | thin, method | [ˈθɪn], [ˈmeθəd] |

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

[1] British English
[2] American English

## Italian phone inventory

| IPA | ETHPA | Example | | IPA | ETHPA | Example | |
|---|---|---|---|---|---|---|---|
| a | a | parete | [paˈreːte] | r͡r | r_r | carro | [ˈkar͡ro] |
| aː | a: | pane | [ˈpaːne] | s | s | salsa | [ˈsalsa] |
| b | b | bambina | [bamˈbiːna] | ʃ | | scena | [ˈʃɛːna] |
| b͡b | b_b | repubblica | [reˈpub͡blika] | ʃ͡ʃ | _ | riuscita | [riuˈʃ͡ʃiːta] |
| d | d | ladina | [laˈdiːna] | s͡s | s_s | deflusso | [deˈflus͡so] |
| d͡d | d_d | freddezza | [freˈd͡dettsa] | t | t | cantata | [kanˈtaːta] |
| d͡dʒ | d_d_ | oggi | [ˈɔd͡dʒi] | t͡ʃ | t_ | cena | [ˈt͡ʃeːna] |
| d͡dz | d_d_z | mezzi | [ˈmɛd͡dzi] | t͡s | t_s | zitto | [ˈt͡sit͡to] |
| dʒ | d_ | Genova | [ˈdʒɛːnova] | t͡t | t_t | viadotto | [viaˈdot͡to] |
| dz | d_z | zona | [ˈdzɔːna] | t͡t͡ʃ | t_t_ | nocciola | [noˈt͡t͡ʃɔːla] |
| ɛ | | mezzo | [ˈmɛd͡dzo] | t͡t͡s | t_t_s | merluzzo | [merˈlut͡t͡so] |
| e | e | terreno | [teˈr͡reːno] | u | u | lumaca | [luˈmaːka] |
| ɛː | : | bene | [ˈbɛːne] | uː | u: | luna | [ˈluːna] |
| eː | e: | nero | [ˈneːro] | v | v | vivace | [viˈvaːt͡ʃe] |
| f | f | fumo | [ˈfuːmo] | v̩v | v_v | provvidenza | [prov̩viˈdɛnt͡sa] |
| f͡f | f_f | caff | [ka ˈf͡fɛ] | z | z | sbarra | [ˈzbar͡ra] |
| g | g | gondola | [ˈgondola] | i̯ | i | inizio | [iˈnit͡t͡si̯o] |
| g͡g | g_g | aggressivo | [ag͡greˈs͡siːvo] | u̯ | u | acqua | [ˈakk͡u̯a] |
| i | i | bilancio | [biˈlant͡ʃo] | | | | |
| iː | i: | lira | [ˈliːra] | | | | |
| ɲ | | gnocco | [ˈɲɔkk͡o] | | | | |
| ɲ͡ɲ | _ | prognosi | [ˈprɔɲ͡ɲozi] | | | | |
| k | k | vacanza | [vaˈkant͡sa] | | | | |
| k͡k | k_k | bocconi | [boˈk͡koːni] | | | | |
| l | l | lama | [ˈlaːma] | | | | |
| ʎ | | figlio | [ˈfiːʎo] | | | | |
| ʎ͡ʎ | _ | bottiglia | [boˈt͡tiʎ͡ʎa] | | | | |
| l͡l | l_l | midollo | [miˈdol͡lo] | | | | |
| m | m | men | [me ˈnu] | | | | |
| m͡m | m_m | mamma | [ˈmam͡ma] | | | | |
| ŋ | | banca | [ˈbaŋka] | | | | |
| n | n | Napoli | [ˈnaːpoli] | | | | |
| n͡n | n_n | nonno | [ˈnɔn͡no] | | | | |
| o | o | posata | [poˈzaːta] | | | | |
| ɔ | | ricordo | [riˈkɔrdo] | | | | |
| ɔː | : | cosa | [ˈkɔːza] | | | | |
| oː | o: | volo | [ˈvoːlo] | | | | |
| p | p | presto | [ˈprɛsto] | | | | |
| p͡p | p_p | scialuppa | [ʃaˈlup͡pa] | | | | |
| r | r | Rimini | [ˈriːmini] | | | | |

## Spanish phone inventory

The second column lists the notations as used in [Lan02].

| IPA | LSch | ETHPA | Example | |
|---|---|---|---|---|
| a | | a | valle | [ˈbaʎe] |
| b | | b | mbar, vino | [ˈambar], [ˈbino] |
| β | ƀ | | cabra, Habana | [ˈkaβra], [aˈβana] |
| d | | d | donde | [ˈdɔnde] |
| | đ | | prado | [ˈpra o] |
| ˛ ( ) | đ ( đ) | _ | juzgado | [xu ˛ ˈɣa( )o] |
| | | | ciudad | [θi̯uˈ a( )] |
| e | | e | pero | [ˈpero] |
| ɛ | | | directo | [diˈrɛkto] |
| f | | f | f cil | [ ˈfaθil] |
| g | | g | gata, tango | [ˈgata], [ˈtaŋgo] |
| ɣ | g̱ | | viga, burgo | [ˈbiɣa], [ˈburɣo] |
| i | | i | pico | [ˈpiko] |
| i̯ | | i | di logo, rey | [ ˈdi̯aloɣo], [ˈr̯ei̯] |
| j̯ | | j | ayer, yuste | [aˈj̯ɛr], [ˈjuste] |
| ɲ | | | a o, ni o | [ ˈaɲo], [ˈniɲo] |
| k | | k | casa | [ˈkasa] |
| l | | l | lejos | [ˈlɛxos] |
| ʎ | | | caballo, llave | [kaˈbaʎo], [ˈʎaβe] |
| m | | m | mismo | [ˈmizmo] |
| n | | n | nunca | [ˈnuŋka] |
| ŋ | | | ancla | [ˈaŋkla] |
| o | | o | toro | [ˈtoro] |
| ɔ | | | ojo | [ˈɔxo] |
| p | | p | padre | [ˈpa re] |
| r | | r | puro | [ˈpuro] |
| r͡r | rr | r_r | torre | [ˈtɔr͡rɛ] |
| s | | s | sala | [ˈsala] |
| t | | t | tomo | [ˈtomo] |
| t͡ʃ | tʃ | t_ | chacho, mucho | [ˈt͡ʃat͡ʃo], [ˈmut͡ʃo] |
| θ | | | cinco | [ˈθiŋko] |
| u | | u | duro | [ˈduro] |
| u̯ | u | u | cueva, cig e a | [ ˈku̯eβa], [θiˈgu̯eŋa] |
| x | | x | Jos , mujer | [xo ˈse], [muˈxɛr] |
| z | | z | isla | [ˈizla] |
| ʔ | | | hoy | [ˈʔɔi̯] |

# D Task Description

Institut für Technische Informatik und Kommunikationsnetze

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

(SA-2015-06)

Task description for the semester thesis

of

Ms. Vanessa Hunziker and Ms. Janine Thoma

Supervisors: Dr. Hui Liang
Dr. Beat Pfister

Issue Date: February 24, 2015
Submission Date: June 5, 2015

## Detecting Strong Prosodic Events

### Introduction

One of the most difficult task of text-to-speech synthesis is prosody generation. Basically, the prosody is largely determined by semantic and pragmatic properties. Such properties can hardly be recognized from the input text, however. A system that has to synthesize speech from arbitrary text can do at the utmost a syntactic analysis of individual sentences. The resulting syntax trees can be used to derive syllable stress levels and prosodic phrases. The syntax-based prosody of the synthesized speech sounds generally quite acceptable.

In some cases, neglecting semantic and pragmatic information may result in wrong prosody, however. Then prosody is not in line with the meaning. For instance, a specific meaning of the following sentence requires to emphasize a certain word. Or the other way round, the meaning of the sentence depends on whether and which word is emphasized.

**My** wife flew to Paris. (means, e.g. not his wife)
My **wife** flew to Paris. (means, e.g. not my daughter)
My wife **flew** to Paris. (means, she did not travel, e.g. by train)

Logically, syntax-based prosody is wrong in these cases. And there is no simple solution to change this for text-to-speech synthesis in general.

In the context of a speech-to-speech translation, however, there is an elegant possibility to solve this problem. We can check whether there is emphasis in the input signal and

which words are actually emphasized. This information can then be used to emphasize the corresponding word of the translated and synthesized output speech.

### Task of this thesis

The task to be solved in the framework of this thesis is to detect salient prosodic events such as emphasis or contrast from the speech signal. Quite often these events are very similar to syllables with a high stress level (sometimes also referred as prominence level), but generally they are stronger. In terms of the physical prosodic parameters this stress means: change of fundamental frequency ($F_0$), lengthened phone duration and increased signal intensity. Sometimes a pause may appear before the emphasized segment of the speech signal. Furthermore, emphasis is known to has an impact on the spectrum of phonemes and thus spectral properties may be useful for the detection of emphasis as well.

Most logically, the detection of strong prosodic events has to rely on prosodic properties, mainly on the $F_0$ contour, the phone durations, the intensity contour and pauses. The task is to extract suitable features from these properties and to construct a suitable classifier (see e.g. [1] or [2]).

Naturally, the semester thesis will start with investigating how prosodic features of emphasis differ from those of its neutral version (and probably of its neighboring words). The relationship between emphases and the prosodic parameters duration, intensity and $F_0$ at different levels as well as and their statistics have to be examined.

It is recommended to work on the multilingual SIWIS speech corpus in which emphases were recorded on purpose. The speakers in this corpus were requested to emphasize a couple of words in a sentence intentionally, and each speaker read 25 sentences with emphasis. The corpus contains German, English, French and Italian recordings. Hence, it will be interesting to see whether/what contributory factors in emphasis are language-independent.

Features for training a classifier to detect strong prosodic events will be selected according to the outcome of the aforementioned first stage. Common classifiers (hidden Markov model, neural network, support vector machine, decision tree, or combinations thereof) should be applied to detecting strong prosodic events. A detailed analysis of the classification results should reveal which classifier architecture performs best. It is recommended to use Matlab.

The work done and the attained results have to be documented in a report (see recommendations [3]) 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] J. Buckow, R. Huber, V. Warnke, A. Batliner, and E. Noeth. Multi-lingual prosodic processing. In *Proceedings of ETRW on Dialogue and Prosody*, Veldhoven (The Netherlands), 1999.

[2] M. Kroul. Automatic detection of emphasized words for performance enhancement of a Czech ASR system. In *Proceedings of SPECOM*, pages 470–473, 2009.

[3] 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).

[4] 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 24, 2015

# References


[1] ETH Speech Processing Group. http://www.tik.ee.ethz.ch/~spr/. [Online; accessed 04-June-2015].

[2] MATLAB feedforwardnet. http://ch.mathworks.com/help/nnet/ref/feedforwardnet.html. [Online; accessed 26-May-2015].

[3] MATLAB Random Forests. http://ch.mathworks.com/help/stats/treebagger.html. [Online; accessed 26-May-2015].

[4] MATLAB Statistics and Machine Learning Toolbox. http://ch.mathworks.com/help/stats/index.html. [Online; accessed 26-May-2015].

[5] MATLAB Support Vector Machines. http://ch.mathworks.com/help/stats/support-vector-machines-svm.html. [Online; accessed 26-May-2015].

[6] SIWIS Project. https://www.idiap.ch/project/siwis/front-page. [Online; accessed 26-May-2015].

[7] L. Brieman and A. Cutler. Random Forests. http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. [Online; accessed 28-May-2015].

[8] J. Buckow, V. Warnke, R. Huber, A. Batliner, E. Nöth, and H. Niemann. Multi-Lingual Prosodic Processing. In *Proceedings of ESCA Workshop on Dialogue and Prosody*, 1999.

[9] T. Ewender, S. Hoffmann, and B. Pfister. Nearly perfect detection of continuous f0 contour and frame classification for tts synthesis. In *Proceedings of Interspeech*, pages 100–103, Brighton (UK), 2009.

[10] M. Kroul. Automatic detection of emphasized words for performance enhancement of a czech asr system. In *Proceedings of 13th International Conference on Speech and Computer (Specom 2009)*, 2009.

[11] E. Nöth, A. Batliner, J. Buckow, R. Huber, V. Warnke, and H. Niemann. A Multilingual Prosody Module in a Speech-to-Speech Translation System. In , editor, *Proc. of the Workshop on Multi-Lingual Speech Communication*, pages 110–115, -, 2000.

[12] B. Pfister and T. Kaufmann. *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer, 2008.