CHALMERS
UNIVERSITY OF TECHNOLOGY

Speech Categorization with Prosodic Features and Deep Learning

Bachelor's thesis in Computer Science and Engineering

DANIEL DAVALLIUS, MARKUS INGVARSSON,
JULIA ORTHEDEN, MARKUS PETTERSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Bachelor's thesis DATX02-19-09

Speech Categorization with Prosodic Features and Deep Learning

Daniel Davallius
Markus Ingvarsson
Julia Ortheden
Markus Pettersson

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2019

Speech Categorization with Prosodic Features and Deep Learning

DANIEL DAVALLIUS, MARKUS INGVARSSON, JULIA ORTHEDEN,
MARKUS PETTERSSON

Supervisor: K V S Prasad, Department of Computer Science and Engineering
Examiner: Wolfgang Ahrendt, Department of Computer Science and Engineering

Bachelor's Thesis DATX02-19-09
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2019
Thesis template provided by David Frisk under the CC BY 4.0 license

Speech Categorization with Prosodic Features and Deep Learning

DANIEL DAVALLIUS, MARKUS INGVARSSON, JULIA ORTHEDEN, MARKUS PETTERSSON
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

The purpose of this thesis is to investigate whether it is possible to perform three separate categorizations of speech based only on pitch and intensity. By using these pitch and intensity curves, the goal is to be able to distinguish between the spoken languages Swedish, Spanish, English, German, French, and Chinese, as well as determining the sex and age group of the speaker, with the use of neural networks.

The pitch and the intensity were extracted from thousands of hours of audio files collected from the Swedish Riksdag and a website with audiobooks in the public domain called LibriVox. When categorizing the age group and the sex, only the audio files from the Swedish Riksdag were used, since they were the only audio files with labels of the sex and the birth year.

The categorization was performed using two different methods. The first was to extract several language characteristic features from the pitch and intensity to use as input data, training multiple feedforward neural networks using the FFNN model, one or more for each categorization. The other method was to use the pitch and the intensity directly as input data to multiple recurrent neural networks using the LFLB-LSTM model, again one or more network for each categorization.

The conclusion is that the LFLB-LSTM model can distinguish between the six languages as well as the sexes and the age groups solely using the pitch and intensity extracted from the audio files. The FFNN model performed significantly worse than the LFLB-LSTM model but still better than pure probability, potentially because of a lack of understanding about what it was in the data that differentiated the categories from one another. Further, it was concluded that it is essential to have sufficient variance in the audio data both within the groups and between the groups. To capture this successfully it is advisable to use sources of audio with a high variance of genders, ages, audio quality, and dialects, preferably by a large number of diverse speakers in each group.

Sammanfattning

Syftet med detta arbete är att undersöka om det är möjligt att utföra tre olika kategoriseringar av mänskligt tal baserat på enbart tonhöjd och intensitet: Utefter språk, utefter kön och utefter åldersgrupp. Inom språk innebär detta att kunna kategorisera mellan sex olika språk: Svenska, engelska, tyska, spanska, franska och kinesiska, såväl som att separat kunna kategorisera talare enligt kön och enligt åldersgrupp enbart baserat på de prosodiska egenskaperna tonhöjd och intensitet.

Tonhöjden och intensiteten extraherades från tusentals timmar av inspelat material som samlades in från Sveriges Riksdag och LibriVox, en hemsida som samlar upphovsrättsfria ljudböcker. Vid kategorisering av kön och ålder användes endast de svenska ljudfilerna från riksdagen eftersom de är de enda som tillhandahåller metadata om detta. Kategoriseringen genomfördes med hjälp av två olika metoder: Den första gick ut på att från tonhöjden och intensiteten ta fram ett antal språkkaraktеristiska egenskaper som sedan används som indata, i träningen av flera feedforward neural networks med en så kallad FFNN-modell. Ett eller fler nätverk användes per kategorisering. Den andra metoden gick ut på att använda tonhöjden och intensiteten direkt som indata till flera recurrent neural networks med en så kallad LFLB-LSTM modell. Även här användes ett eller flera nätverk per kategorisering.

Slutsatsen är att LFLB-LSTM-modellen kan skilja på de sex språken, könen och åldersgrupperna baserat enbart på tonhöjden och intensiteten som extraherades från ljudfilerna. FFNN-modellen presterade signifikant sämre än LFLB-LSTM-modellen, men fortfarande bättre än ren slump. Potentiellt beror detta på en brist på förståelse angående vad i datan det var som skilde kategorierna åt. Vidare så drogs slutsatsen att det är viktigt att ha tillräcklig varians av ljudfiler både inom grupperna och mellan grupperna. För att uppnå detta så är en normal distribution inom datan i form av kön, ålder, inspelningsutrustning och dialekter av hög prioritet, inspelad av en representativ grupp människor.

Keywords: Deep Learning, Prosodic Features, Spoken language recognition, LSTM, CNN, Local Feature Learning Batches

Acknowledgements

Primarily we would like to thank our supervisor K V S Prasad for his support and endless ideas. Further, Morteza Haghir Chehreghani has been a great resource when it comes to feedback and questions regarding the implementation of the artificial neural networks. Finally, we would also like to thank Hari och Maitane for recording audio files for us during the creation of our first prototype.

List of Acronyms

ANN Artificial neural network.
API Application programming interface.

CNN Convolutional neural network.

FFNN Feedforward neural network.

LFLB Local feature learning block.
LSTM Long short-term memory.

MAR Missing at random.
ML Machine learning.

NaN Not a number.

RNN Recurrent neural network.

SDK Software development kit.

Contents

1
Introduction

Even when disregarding the specific words, languages can vary significantly from one another and can tell quite a lot about a person. Some languages are known for being melodic while others are known for their rapid pace or their sharp sounds. To make it even more complicated, the way of speech can vary by accents or speaker-specific features as well, possibly due to the sex or the age of the speaker. As the voice changes with human age, it should be possible to extinguish things like changes in the pace of the speech and the loudness of the speech due to hearing problems. There are also studies that claim that the male pitch rises, and the female pitch slightly decreases with age (Butler, Lind, & Weelden, 2013). Further, since the fundamental frequency generally differs between male and female speakers, the sex should also be recognizable.

The ability to categorize speech, for instance between languages, can be useful for many applications such as multilingual speech recognition, translation, call center optimization, and automatic data labeling (Pi school, 2017). Categorization is a practice that has been around for a long time. Even Aristotle wrote about it in his text *Categories* in the collection *Organon*. A lot has happened during the last two millennia when it comes to creating models for categorization. Different strategies exist for classification, where the use of artificial neural networks (ANN) has grown in popularity the recent years. The overall idea of an ANN is to create a synthesized model of the brain, where similarly to the brain, the model is created by training the network that a particular input should map to a specific output. A robust architecture is required when one builds an ANN, which has to support large amounts of data, and potentially other features as well. Synthetic memory would be such a feature, which works well with sequential streams of data over time, such as when one is looking at audio files. Neural networks are useful in pattern recognition, where it has shown to be useful in interpreting speech (Graves, rahman Mohamed, & Hinton, 2013).

1.1 Purpose

The project aims to make use of the prosodic features pitch and intensity, extracted from audio files, to perform various categorizations of human speech with the help of neural networks. By using data of the pitch and intensity values over time, the goal is to be able to recognize the following different characteristics of the speaker:

- Determining what language is being spoken, choosing from English, French, Spanish, German, Chinese or Swedish.
- Determining the sex of the speaker, choosing from male and female.
- Determining the age group of the speaker, choosing from speakers born before 1955 and speakers born after 1975.

Two different models of neural networks are implemented and evaluated. The first model extracts specific features from the pitch and intensity of the audio files and uses this as input data to train a feedforward neural network (FFNN). The second method uses the unmodified data of pitch and intensity when training a local feature learning block - long short term memory (LFLB-LSTM) network. In both methods, the models are created per category of prediction and are compared to one another.

The purpose is to investigate whether it is possible to identify the mentioned categories of a speaker solely based on these prosodic features and what conclusions can be drawn from categorizing these attributes through the neural networks. Identifying is in this case defined as the probability of success of the predictions being higher than if done at random.

1.2 Scope

The project focuses on the prosody of the human voice. Word interpretation is entirely excluded, as are other ways of understanding intent such as body language, facial expressions, or context. Within the field of prosody, the focus is solely on certain features, namely pitch and intensity. The focus is on categorization between groups, meaning the ability to distinguish between groups speaking the languages German, English, Spanish, French, Swedish and Chinese, as well as between groups divided by their sex or their age.

The neural networks are trained with labeled data, and the network is implemented through a preexisting Python library called Keras. The audio data is gathered from a public domain audiobook website called LibriVox, and via an external API for the Swedish Riksdag, where interpellations have been used.

2
Theory

Speech categorization has been of interest for many years (Vicsi & Szaszák, 2010), and researchers have tried to learn more about prosody to improve automatic speech recognition. Artificial neural networks have shown to be useful in categorizing speech (Graves et al., 2013), where hidden Markov Models or k-nearest neighbor algorithms (de Bruin & du Preez, 1993) has done the job in the past. In this section, the basics of prosody, along with background information about artificial neural networks and related research, will be presented.

2.1 Prosody

Prosody is the study of how the meaning of speech is affected by its tune and rhythm over time (Manell, 2008a). Prosodic information plays a vital role in human speech communication. The information can change the meaning behind a phrase due to vocal changes, where a few examples are irony, sarcasm, or statements vs. questions. Prosody at an acoustic level is primarily characterized by the vocal pitch, loudness, and rhythm.

2.1.1 Pitch

Pitch is closely related to the frequency of vibration of the vocal cords, and it varies from person to person (Vajda, 2001). The pitch can be higher or lower, depending on the speaker's age, sex, or language. The variation of pitch is called intonation. The pitch variation helps to characterize the prosody and is a variable that is used to understand the meaning of the words for some languages, such as Mandarin and Cantonese (Oxenham, 2012). Pitch is also something that helps a listener to isolate a speaker when multiple conversations are happening in parallel. A common way to recall a pitch in speech is through a complex harmonic tone, which is analogous to the fundamental frequency, measured in Hertz (Hz).

2.1.2 Sound intensity and rhythm

Sound intensity can be described as sound pressure. It is the measure of changes in air pressure that one experience as sound, measured in decibels (dB). A way to represent this is by using a two-dimensional waveform, which presents the time domain representations of the intensity variation over time (Manell, 2008b). Intensity is the physical variation of sound pressure, while loudness is a perceptual construct.

Through a waveform representation, one is also able to view the rhythm of the sound, which in this case is the alternations of the intensity over time (Gibbon, 2017). Rhythm is something that varies from language to language due to different styles of syllable use and phrase structure.

2.1.3 Stress-timed and syllable-timed languages

The term rhythm can be divided into subcategories. By comparing multiple languages, one can see distinctions between the rhythm, where some languages are syllable-timed, and some are so-called stress-timed languages (Conlen, 2016). A syllable is a single unit of speech, often containing a vowel, and stress is the emphasize of a syllable. The first person who developed these two subcategories of rhythm was Lloyd James, an Australian linguist who compared the rhythm of Spanish with the sound of a machine gun. Spanish is said to be a syllable-timed language, while English is a stress-timed language, which James thought had similarities with Morse code.

In Spanish, the syllables last the same amount of time and are not dependent on whether the syllable is stressed or not. In English, however, many syllables get shortened depending on where the stress lays in the sentence. An example is the word "America," where the emphasis of the second syllable makes it sound longer compared to the rest of the syllables. Other syllable-timed languages are Italian, French, and Chinese (Mok, 2009), while stressed-timed languages also include German (British Council, n.d.), Swedish (Frankfurt International School, n.d.), Dutch (Collins & Mees, 1984). Portuguese belongs to both categories, where Brazilian Portuguese is classified as syllable-timed and European Portuguese as stress-timed (School, n.d.).

2.2 Physical properties of prosody

Not all properties of prosody are objective physical properties (Hartmann, 1997). Some are also considered to be psycho-acoustical attributes of sound. They are perceived differently depending on the percipient. This fact can complicate things when one is analyzing and studying prosody for apparent reasons. The question arises whether a difference between sounds contains any actual physical differences or if it is just a change one's experience of the sound.

Intensity is a physical property that is measured by the power carried by sound waves, divided by the area. Its perceptual representation mostly corresponds to loudness, see figure 2.1 (Wolfe, 2019).

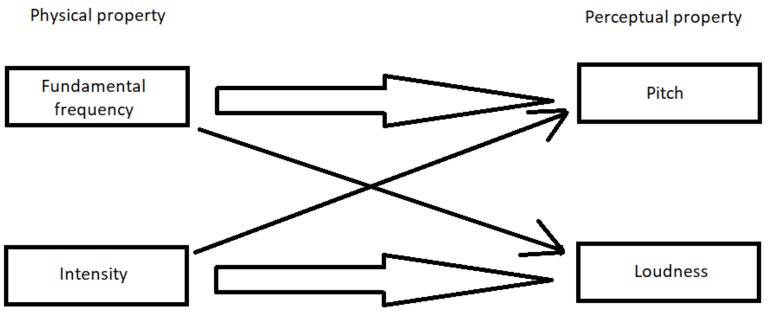


Figure 2.1: An illustration of how the physical properties of fundamental frequency and intensity are correlated with the perceptual properties pitch and loudness. The pitch correlates mainly with the fundamental frequency, but also slightly with the intensity. Loudness correlates mainly with the intensity, but also slightly with the fundamental frequency.

One of the physical attributes of sound that gets produced when a person is speaking is the frequency of the vocal cords, also referred to as the fundamental frequency (Li & Jain, 2009). Pitch is usually referred to as how one perceives the fundamental frequency (Wolfe, 2019). Pitch correlates with frequency, and by doubling the frequency, the pitch will increase by an octave. Other factors come into account when looking at the pitch variance, but the frequency is a good indicator of whether the pitch will rise or fall. See figure 2.1.

2.3 Artificial neural networks

Artificial neural networks (ANNs) are a form of self-learning algorithms used in machines to imitate the workings of an anatomical brain (Mehlig, 2019). They are made up of artificial neurons, which are greatly simplified versions of biological neurons. An artificial neuron takes an input, multiplies it with a weight and yields an output through an activation function (see figure 2.2). The network consists of layers of these neurons. The first one is called the input layer, after that comes one or more hidden layers and lastly the output layer. Adding layers makes the network deeper and allows the network to recognize more complex patterns.

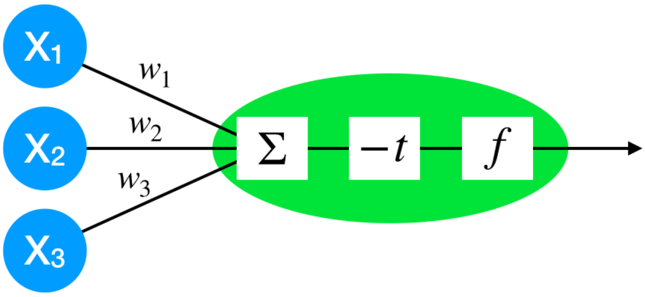


Figure 2.2: Illustration of an artificial neuron with three inputs. Each input, x_n, has a corresponding weight, w_n, with which it will be multiplied. The products from these operations will be summed up and the threshold t will be subtracted. It is these variables, w_n and t, that are updated during the learning process. The resulting value will be passed through a simple activation function f, such as *tanh* or the sigmoid function. The output of the neuron will therefore be equal to $f(w_1x_1 + w_2x_2 + w_3x_3 - t)$

Artificial Neural networks can learn to recognize structures and patterns in a dataset by updating the weights of its neurons (Mehlig, 2019). This training process is executed by inputting data, for which the target output is already known, into the ANN and updating its structure until its output is satisfactory. The network can then apply this knowledge to make estimations for new input data, where the target values are not known beforehand. Neural networks have a broad scope of use cases in both business and academia, where the aim is to find non-obvious correlations for a set of data.

2.3.1 Feedforward neural networks

One of the implementations of an ANN is the Feedforward neural network (FFNN). The design is a rather straightforward implementation, without any of the more intricate features of more advanced architectures (Mehlig, 2019). It merely consists of layers with regular neurons where all neurons in one layer have a connection to all of the neurons in the next one (see figure 2.3). Hence, these layers are said to be a fully connected layer.

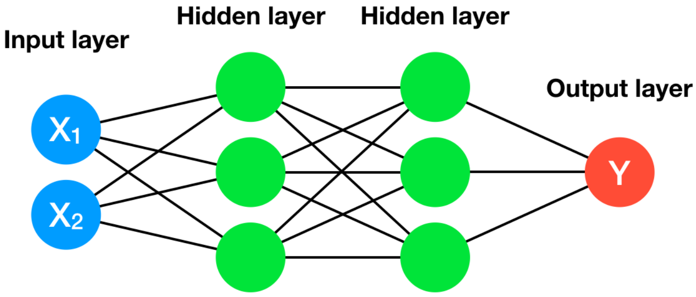


Figure 2.3: Illustration of a simple, deep FFNN. Two inputs, X_1 and X_2, result in an output Y after being passed through and processed by the neurons in the two hidden layers.

2.3.2 Recurrent Neural Networks

The distinguishing feature of a Recurrent Neural Network (RNN) is the fact that it preserves information about earlier inputs (Sherstinsky, 2018). A recurrent node will have its output as an input, which means that the information calculated from the previous input will be taken into account when making the next calculation. This feature is useful when the incoming data is structured in a sequence where the context of the input is significant for its meaning. An obvious example would be sentences, where a word can mean different things depending on the other words surrounding it. It is often easier to visualize an RNN as "unfolded", where each time step is presented as a separate layer (see figure 2.4). In the last few years, there have been plenty of successful implementations based on these kinds of networks (Graves et al., 2009; Li & Wu, 2014). In its purest form, an RNN struggles to pick up on long-term dependencies since events that occurred a long time ago are quickly forgotten. More advanced versions of the RNN architecture have been developed to address these issues (Hochreiter & Schmidhuber, 1997).

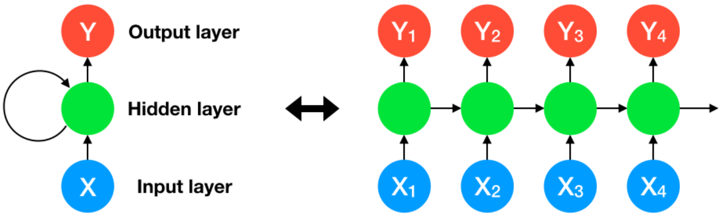


Figure 2.4: Illustration of a simple RNN. The model consists of a single hidden layer with a single recurrent neuron. To the right is a representation of the same network unfolded. If the input X is a sentence each step would represent processing a new word (X_1, X_2, ..., X_n). A new output (Y_1, Y_2, ..., Y_n) is also given at every step. The output could for example be the estimated mood of the writer.

Long Short-Term Memory networks

The Long Short-Term Memory (LSTM) network is one of the most commonly used versions of the RNN architecture. It has provided excellent results for many different applications, such as speech recognition and intent analysis (Graves, Mohamed, & Hinton, 2013; Zyner, Worrall, Ward, & Nebot, 2017). The main advantage of LSTM architecture is its ability to learn long-term dependencies very well. This trait is achieved by using a so-called cell state, which is a connection that runs along the entire unfolded chain of time nodes (Hochreiter & Schmidhuber, 1997). The cell state retrieves the output from each time node in a way that makes it very easy for information to flow unaltered down along the unfolded chain (see figure 2.5). This way allows for the network to "remember" things in a way which is practically impossible for a regular RNN.

As an example, it is possible to train an RNN to fill in the blanks in a sentence like "The capital of Italy is _____." It would, however, be impossible to do the same for a sentence like "We arrived in Italy by ferry and had to rent a car in order to drive to the country's capital, _____". This result is due to the long distance between the blank space and the critical word, "Italy." An LSTM network would be able to learn both.

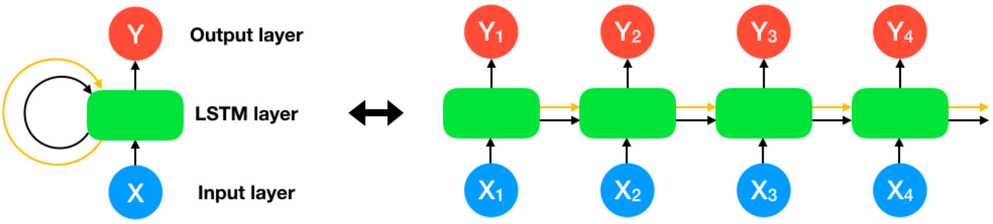


Figure 2.5: Illustration of an LSTM network. The main difference compared to an RNN is the addition of a cell state, represented with a yellow arrow. This cell state runs along the entirety of the unfolded model and allows the network to remember events that occurred much earlier in the chain.

2.3.3 Convolutional Neural Networks

Convolutional Neural Networks are a category of Neural Networks that have been successfully used mostly in image recognition and classification. Convolutional Neural Networks are made up of neurons that each have learnable weights and biases. Each neuron receives an input, performs a dot multiplication, and optionally follows it with a non-linear computation. The significant difference compared to regular FFNNs is the increased efficiency in the forward function and the reduced amount of parameters due to the assumption that an image is used as input data.

A convolutional neural network consists of four building blocks: A convolutional layer, a non-linearity layer, a pooling layer, and a fully-connected layer (Albawi, Mohammed, & Al-Zawi, 2017).

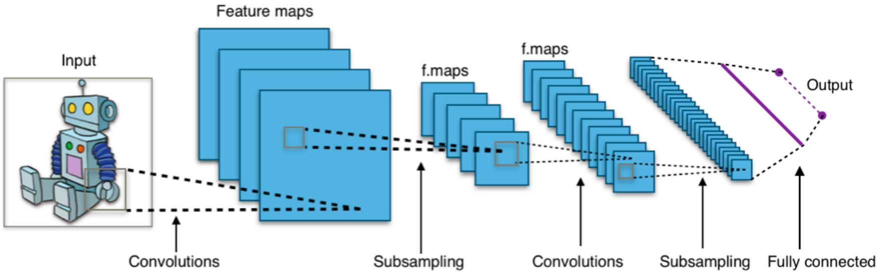


Figure 2.6: An illustration of the flow of a common CNN. (Aphex34, 2015).

Convolution is the process in which smaller regions of the input image are processed to learn image-specific features, as Albawi, Mohammed, and Al-Zawi explain it. These features are learned by using different filters, which are methods of computations when processing the image regions in order to generate a feature map of the whole image. In order to map these regions, a so-called kernel is used, with the same dimensions as the regions that should be mapped. This kernel will iterate over the entire input with a fixed step size denominated as its stride (see figure 2.7 for an example).

An advantage of using ANN compared to linear functions is the level of complexity that can be described. The most common way to ensure this non-linearity is to use an activation function called the rectified linear unit (ReLU) (Yamashita, Nishio, Do, & Togashi, 2018). ReLU is computed element-wise to replace the negative values with zero and introduce non-linearity in the feature map. On the processed feature map, a pooling method enables a reduction of the complexity by calculating a value that is representative of the region of the image. One example is Max Pooling, in which the most significant element of each region is extracted to form a new output. The output is then classified between several categories using the last building block, the classification.

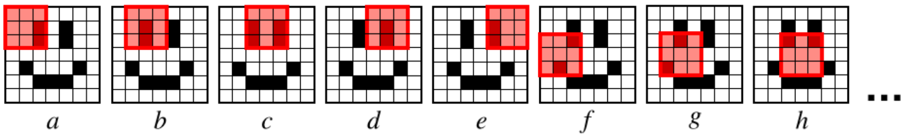


Figure 2.7: A visualization of the feature mapping in a 2D convolutional layer. The kernel has a size of 3×3, a horizontal stride of 1 and a vertical stride of 2. Note that the feature map for b and d are given the exact same input and will therefore be recognised as equal.

Although most commonly used for working with two-dimensional image data, CNN's have also been successfully deployed on one-dimensional time series, such as ECG for heart monitoring (Kiranyaz, Ince, Hamila, & Gabbouj, 2015) and vibration data to predict structural damage on buildings (Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017). These work just like the more conventional 2D CNN's, with the only difference being the dimensionality of the input.

2.3.4 Local Feature Learning Blocks

The Local Feature Learning Block (LFLB) is a standalone collection of ANN layers (see figure 2.8) directly derived from the CNN architecture (Zhao, Mao, & Chen, 2019). It consists of a convolutional layer, a batch normalization layer, an exponential linear unit, and a max pooling layer. The convolutional layer, the linear unit, and the pooling layer work in the same way as in a CNN. The batch normalization layer is added to increase the rate at which the network learns (Ioffe & Szegedy, 2015).

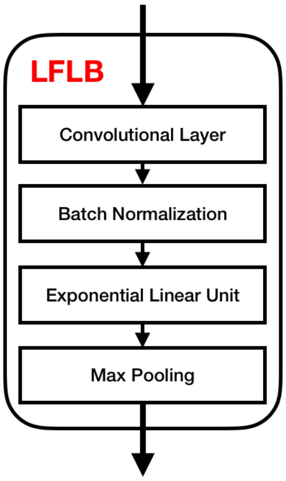


Figure 2.8: The internal architecture of a Local Feature Learning Block (LFLB).

The purpose of this block is to recognize local features in either a 2D matrix, like a spectrogram, or a 1D array, like a velocity vs. time graph. The blocks can be stacked on top of one another to recognize more complex and abstract features.

An LFLB has the following parameters:

- **Filters**: The number of different features that should be mapped in the convolutional layer. Adding too many filters may lead to over-fitting, but too few will result in the block oversimplifying crucial information.

- **Kernel size**: The dimensions of the features that should be mapped in the convolutional layer. See figure 2.7 for an example.

- **Padding**: In order to ensure that the output from the convolutional layer has a specific dimension, some form off padding might be added to the input. Often the output should have the same shape as the input, which can be achieved by adding padding of half the kernel size around the input. See figure 2.9 for an example.

- **Strides**: The step size between each feature map. See figure 2.7 for an example.

- **Pool size**: An LFLB only returns the highest scoring feature map within a region as its output, just like in the pooling layer of a CNN. This value is obtained through the max pooling process. Pool size determines how large these regions should be, in other words, how much smaller the LFLBs output should be than the input.

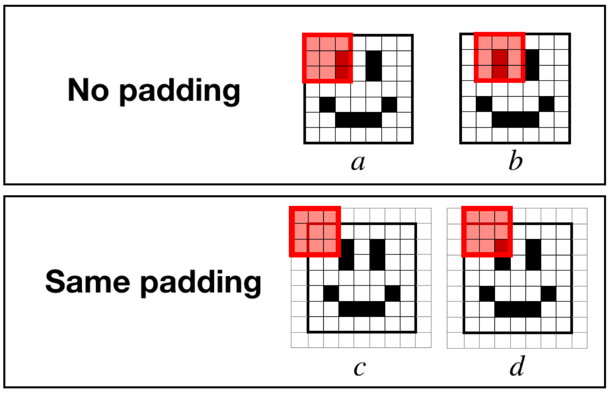


Figure 2.9: The first two feature maps for a 7 × 7 figure with and without same padding (kernel size 3 × 3, horizontal stride (1, 1)). The output shape without any padding will be 5 × 5. With padding the output shape will be 7 × 7, the same as the input.

2.3.5 Training aspects

The training of the artificial neural network is an essential part of the implementation with several terms and techniques involved. This section will explain the critical concepts of how to train the models.

Hyperparameters

In ANN, not all parameters can be learned. Some have to be set and tweaked manually, usually before the training begins. These parameters express the higher-level properties of the model, such as its complexity. There is often no right or wrong answer regarding how to set these hyperparameters. Instead, the optimal values are found by testing different values and different models. It can often be useful to look at similar projects and the architecture they have implemented.

Deciding the number of neurons in the hidden layers is an essential part of deciding the architecture of the ANN (Smith, 2018). If this number is too low, it results in something called underfitting. Underfitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated dataset.

Using too many neurons is equally problematic since it can cause overfitting, which means that the neural network through its higher complexity memorizes not only the relevant properties but also the unrelated properties of the dataset such as irrelevant noise (Yamashita et al., 2018). This structure can create a model that is very suited to fit the dataset it was trained to fit and will consequently struggle to categorize new data. Therefore, finding a balance in the number of neurons of the hidden layers is time-consuming but crucial.

It is essential to have qualitative data to learn from to achieve a good result, and it is equally essential to have a validation set to analyze the performance of the network frequently. The validation set has to be completely separate from the training dataset to detect any overfitting in the training set.

The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset. Each epoch gives the network a chance to tune the parameters according to the data. Normalization is performed on the data before using it for training, to ensure that each feature is of equal importance. Commonly, values between zero and one are used for all the training data. One way to do this is to subtract the average and then divide by the range for all the training data. The batch size is the number of samples processed before the parameters of the model are updated. The predictions are compared at the end of the batch to the expected output variables, and an error is calculated. From this error, the update algorithm is used to improve the model, e.g., move down along the error gradient.

Another hyperparameter to consider is the kind of activation function to use when training an ANN. The activation function is uniquely specified for each layer, so this consideration will have to be made for each one of them. There are two different types of activation functions, linear and non-linear. The choice of activation function depends on the problem at hand and to some part, personal preferences (Konstantin-Klemens, 2018). Konstantin-Klemens continues to explain that when determining the update of the weights, the gradients of the activation function for each node in the network is computed. The choice of the activation function is essential for each hidden layer not to compute too small or too large gradients, which would prevent the neural network from improving. Some examples of standard non-linear activation functions are ReLu, Softmax, and Sigmoid.

Regularization

The process of reducing over-fitting while maintaining the size of the network is called regularization. Regularization reduces over-fitting by adding a penalty to the loss function. This penalty ensures that the network does not learn individual features from the dataset. The two main regularization techniques used in this project will be explained in the following section.

Dropout is an approach that minimizes the interdependent learning among the neurons and therefore, forces the network to learn more robust features instead. To do this, it randomly drops some output nodes, which makes the layers behave like they have different numbers of nodes (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Early stopping is another alternative to reduce the risk of over-fitting by establishing a limit for the number of times the network can run without improvement. If the network exceeds this limit, it stops the training since continuing would likely result in a bias to the training set.

2.4 Related research

Language recognition through the use of prosody has been studied before, one study, in particular, was found to be similar to the aim of this project. In 1993, a group of researchers at the University of Stellenbosch, South Africa who were working on signal processing, attempted to classify speech in the three languages Afrikaans, English, and Xhosa (de Bruin & du Preez, 1993).

In the study, they extracted certain features from a "pitch contour," which they generated from the speech. They then looked at these features and tried to distinguish between the languages. They discovered that not all languages were equally recognizable. Although the researchers of Stellenbosch could easily classify Xhosa and Afrikaans with 89% accuracy, they were not as successful when processing English. Furthermore, the findings suggest that the mother-tongue of a speaker influences their prosody when speaking another language. The result may have been affected by the fact that non-native speakers may have recorded a significant amount of English recording. In the study, they also remark: "Results indicated an excellent distinction between a tone and a stress language, Xhosa and Afrikaans." This outcome may indicate that it would be easier to tell two languages apart in this manner if they have distinctive linguistic characteristics.

Researchers have made progress when it comes to speech in artificial neural networks. In 2013 a group of researchers from the department of computer science of the University of Toronto trained a recurrent neural network in speech recognition. (Graves et al., 2013). The traditional way to create the network was to combine it with other models, such as the commonly used hidden Markov model. (Kamble, 2016).

In the study at the University of Toronto however, the network was trained by using end-to-end training, where the network picks up how to map straight from acoustic to phonetic sequences, and a Long Short-Term Memory architecture. They achieved a test set error of 17.7%, a quite satisfactory result.

The idea to use a recurrent neural network for speech recognition is not a new one, however. In the University of Cambridge T. L. Burrows and M. Niranjan wrote in a paper, back in 1994, about the different areas of use with RNNs for classification, where speech recognition was specifically mentioned (Burrows & Niranjan, 1994).

3
Method

This section describes how the project was structured and how the two different models were created: the LFLB-LSTM model and the FFNN model. First, there will be an introduction to how the data was collected and preprocessed, followed by a description of the neural networks and how they were optimized and evaluated. Finally, there will be an overview of the tools used in this project.

3.1 Data collection

Several hundred hours of audio files were needed to train the models and generate satisfying results. Since no readily labeled dataset that met the requirements for the project was found, data collection and correct labeling became a crucial part of the project.

3.1.1 Requirements on audio

Certain desirable features were decided upon to ensure the quality of the collected audio data:

- In order to notice differences between various speakers, files with only one speaker each was set up as a requirement.
- In order to be able to train a neural network to recognize differences between languages, audio files containing more than one language were avoided.
- In order to have the neural networks pick up the prosody of the speaker, background noise was preferred to be kept to a minimum.
- In order to enable the extraction of prosodic features, the audio files needed to be at least a couple of sentences each.
- In order to draw any conclusions about our given categorizes, a sufficient amount of data was required. This goal was set to at least a hundred hours of audio per category.

The copyright needed to be considered even with a source of audio files that fulfilled all the requirements stated above. Copyright infringement can be avoided by only using public domain sources or by getting permission to use a particular dataset.

3.1.2 LibriVox

LibriVox is a public domain online depository which hosts audiobooks, where volunteers upload recordings of works in the public domain (*LibriVox*, n.d.). The majority of books are recorded in English, but several other languages are available as well. The site offers an open API for downloading audio files as well as metadata. A script was created to allow filtering recordings by language, which found the desired audiobooks and downloaded the mp3 audio files and related metadata.

3.1.3 The Swedish Riksdag

The Swedish Riksdag has made a public API available for accessing resources concerning members of the Riksdag, voting records, and plenary sessions (*Dokumentation - Riksdagens öppna data*, n.d.). Among these resources are audio recordings for all speeches made in the Riksdag since 2001 as well as time stamps for when a given speaker was speaking.

The Riksdag also has an API for fetching information about the speakers themselves. Out of a platitude of available information, sex, year of birth, and year of recording were stored alongside the prosodic information for each speech. A script was created for downloading the audio files alongside the metadata, processing them to extract pitch and intensity before uploading the information to a database. There ended up being 23403 such recordings in the database by 425 different speakers from 2540 Riksdag interpellations.

3.2 Audio preprocessing

The neural network required a consistent shape of the data for every source file. The data that was extracted from each audio file was one array of pitch representing the value of the fundamental frequency over time and one array of intensity representing the value of the amplitude over time.

The functions "to_pitch" and "to_intensity" of the Python library Parselmouth were used, which returned the pitch and intensity objects containing the desired arrays. See section 3.7.3 for further details. These arrays were then ready to be used as input data for the neural networks.

3.3 LFLB-LSTM model

Unlike the FFNN method, the only preprocessing necessary for this method was to extract the speaker information, the pitch array, and the intensity array. An architecture inspired by the one presented in the paper *Speech emotion recognition using deep 1D & 2D CNN LSTM networks* was used as the basis for this model (Zhao et al., 2019). Zhao, Mao, and Zhen implemented an ANN that categorized an unprocessed audio signal based on the speaker's emotional state.

The inner workings of deep neural networks are notoriously difficult to understand (Lipton, 2016), but some speculative motives for why this architecture has been used will be provided alongside the description of the model. Whether the model behaves as intended or if it finds its unique way to solve the problem is unknown. That being said, the model consists of three sequential parts which each have distinct responsibilities, presented in the rest of this section.

3.3.1 Local feature component

The first part was responsible for finding spatial features in the input signal. The idea was that this part of the network would learn to recognize slopes, peaks, and other local characteristics of the pitch and intensity curves. The input signals would then be partitioned into smaller parts, which would each be categorized as the closest matching local feature. On a higher level, this would equate to recognizing local prosodic features such as intonation, stress, and pauses (Vaissière, 1983). Much of the complexity was removed from the unprocessed pitch and intensity curves by simplifying the two signals to an idealized sequence of categories, which should be easier to classify for the neural network.. This architecture was achieved with four local feature learning blocks (LFLBs), which specialized in finding and classifying these local correlations (Zhao et al., 2019).

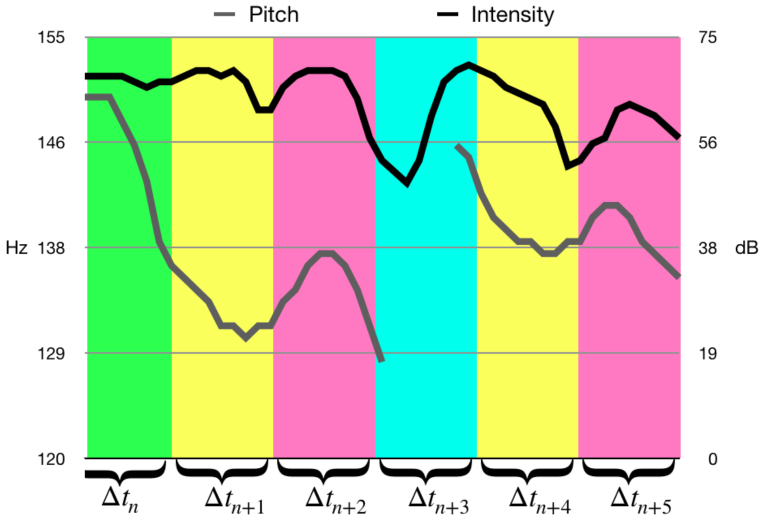


Figure 3.1: A toy example of the output from the first LFLB. Each time span (Δt) has been assigned one out of a fixed set of categories correlating to the shape of the pitch and intensity curves. Each category is represented with a color. In this example, Δt_{n+1} and Δt_{n+4} belong to the same category, since they both have a falling pitch and a falling intensity. Δt_{n+2} and Δt_{n+5} are also grouped since they both contain peaks. There are no guarantees that the trained model makes these classifications

3.3.2 Global feature component

An LSTM layer was appended to the model to discover long-term dependencies in this sequence of local features. This layer specializes in extracting information from time series and have in the past shown great results for finding correlations between different parts of an input signal which are far apart (Hochreiter & Schmidhuber, 1997). The intention was that the network would pick up on what aspects of these local feature series distinguished each category.

3.3.3 Fully connected layer

A fully connected layer was added at the end of the model to get the output in a way that was suitable for categorization. This layer outputted a vector with a score for each of the categories. The higher the score, the more certain the network was that the current sample belonged to that specific category.

The pitch sequence and the intonation sequence that was being generated from the script to convert the audio files were what was being used as input data in this implementation. The language, transformed into a one-importance matrix was the output data.
Figure 3.2 shows the final architecture of the LFLB-LSTM model.

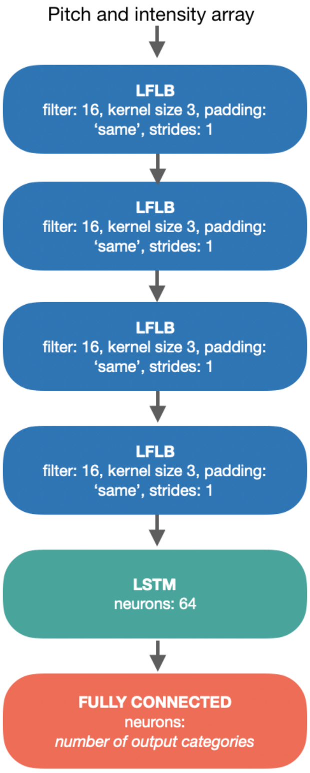

Pitch and intensity array
LFLB
filter: 16, kernel size 3, padding: 'same', strides: 1
LFLB
filter: 16, kernel size 3, padding: 'same', strides: 1
LFLB
filter: 16, kernel size 3, padding: 'same', strides: 1
LFLB
filter: 16, kernel size 3, padding: 'same', strides: 1
LSTM
neurons: 64
FULLY CONNECTED
neurons:
number of output categories

Figure 3.2: The final architecture of the LFLB-LSTM model, each LFLB layer consists of first a batch normalization, thereafter an ELU operation and lastly a Max Pooling1D layer with pool size 4 and strides 4.

3.4 Feedforward neural network model

This model was inspired by the feature extraction used in the South African study "Automatic Language Recognition based on Discriminating Features in Pitch Contours" (de Bruin & du Preez, 1993). In the study, prosodic features distinguishable from the pitch contours were extracted from three different categories: intonation, tone, and duration. The features in the intonation and duration categories were calculated on full sentences while the ones in the tone category only handled word specific calculations.

Only the features concerning intonation were extracted in this project. Additional features concerning the average and the variance of the intensity were added later since similar patterns could be seen in the intensity plots as in the pitch plots. In addition to the intonation features described above, the choice was made to add four features concerning the pauses, stated in table 3.1. The extraction of the features was then implemented in a Python script through which the audio files were processed one by one after being divided into 16-second clips. The rest of this section describes in more detail the features extracted in the respective category.

Table 3.1: Features capturing the duration of speech/silences in the pitch and intensity sequences.

Feature description capturing the duration of speech/ silence
Average length of speech
Average length of silence
Variance of length of speech
Variance of length of silence

3.4.1 Features Concerning Intonation

Both the rate of the speech and the duration of the speech are considered to provide language-specific features concerning the intonation. In the study "Automatic Language Recognition based on Discriminating Features in Pitch Contours", they illustrate this by an example: "if the speech rate is very slow, the number of fluctuations (that is, the total number of positive and negative slopes) in the sentence may not be typical of the language". They continue by stating that to ensure it is as language specific as possible, it is not enough to treat different aspects from each category. One also needs a large set of speakers. Therefore the dataset consists of 1191 unique speakers in total.

All intonation features used are listed in table 3.2 and are extracted from the 16-second audio clips.

Table 3.2: Features capturing the intonation that were extracted from the pitch contour and intensity contour.

Feature description capturing intonation in a sentence
Variance of pitch
Average gradient of positive pitch slopes
Average gradient of negative pitch slopes
Total number of positive pitch slopes
Total number of negative pitch slopes
Average length of positive pitch slopes
Average length of negative pitch slopes
Average of intensity
Variance of intensity

3.4.2 Implementation of feedforward neural network model

The 14 language characteristic features extracted from the audio files (2 for intensity and 12 for pitch) were used as input training data and were later fed into the network in batches of 128. The loss was calculated for each batch and the network parameters updated. The architecture of the resulting network was determined by comparing three architectures used for a similar purpose and choosing the highest performing one (Mary & Yegnanarayana, 2008; Leena, Srinivasa Rao, & Yegnanarayana, 2005; Mary & Yegnanarayana, 2008). The resulting architecture is shown in figure 3.3. The first layer uses ReLu as activation function, and the rest use the activation function Softmax. All layers are fully connected layers. The data is shuffled, and then divided into a training set, validation set, and test set. The training set consists of 80% of the total amount of data, the validation and the test set contains 10% each. The training data is randomly initialized for each training of a neural network. It is guaranteed that the same speaker cannot appear in all of the three sets. The language, transformed into a one-importance matrix is the output data.

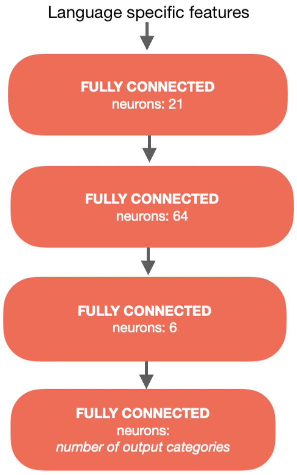


Figure 3.3: An illustration of the final architecture of the feature model. The first layer uses relu as activation function and the other layers use softmax. The input consists of language specific features extracted from the intensity and pitch array.

3.5 Optimization

From the start, the simplest forms of architecture for both models were implemented, with only a single hidden layer. The models were later optimized as the number of categories to predict was increased.

Due to time limitations, only a few tests were done to compare different architectures for the models. The architectures tested were based on reports with similar objectives. After that, the number of hidden layers, as well as the number of neurons, were tweaked. The architectures that achieved the best results were then used to generate the results of the two models.

3.6 Evaluation

The training data was divided into a training set, a validation set, and a test set each time a new model was trained. The validation set was used for regular evaluation and to tune the hyperparameters after each epoch. The purpose of the test set was to evaluate the final model after the training was completed. The test set consisted of entirely new audio files that have not been used during training to ensure an unbiased final evaluation of the model.

Confusion matrices were being used to evaluate the performance of the models further. A confusion matrix shows the number of false predictions as well as the number of true predictions for each category. This matrix, in combination with the total accuracy of the model and the behavior of the loss function of the model, provided a basis to determine how well the model was performing.

Another evaluation method was to insert files that were not used when training the model or were not retrieved from LibriVox or the Swedish Riksdag to see how well the model performed. Further, audio files from other languages (Dutch, Italian, Japanese, and Portuguese) were inserted into the neural network to analyze the output. Other metrics that were considered were the number of categories/languages being classified and the number of features the different models could label.

3.7 Tools

The most important tools used in the project are described more thoroughly in this section.

3.7.1 Python

All parts of the program were implemented in Python, which is a high-level, general-purpose programming language that is commonly used in Machine Learning projects (*The Python Programming Language*, n.d.). This choice was based on the many libraries available, which ease the computations and enable integration with the other tools mentioned above.

3.7.2 Keras

Keras is a high-level neural network API, written in Python. It was developed with a focus on enabling fast experimentation (Chollet et al., 2015). In Keras, models are used to organize the different layers of the neural networks. Changing between various architectures and tweaking hyperparameters is very easily done. In this project, Keras was used for the implementation of the various neural networks, primarily because the group had previous knowledge of the program but also because of advice from Morteza Haghir Chehreghani, associate professor at Chalmers.

3.7.3 Parselmouth

In order to extract the information that was desired from audio files (the pitch and the intensity), the python library Parselmouth was used. It implements functions found in the audio engineering application Praat (Jadoul, Thompson, & de Boer, 2018; *Parselmouth – Praat in Python, the Pythonic way*, n.d.). It has the capability of reading audio files, and manipulating the read contents, for example by extracting the fundamental frequency as a function of time, or the intensity of the sound as a function of time.

In order to extract the pitch, more specifically the fundamental frequency, Parselmouth uses an algorithm which "performs an acoustic periodicity detection on the basis of an accurate autocorrelation method" (*Praat - Sound: To Pitch (ac)...*, n.d.; Boersma, 1993). The resulting "pitch object" contains an array with the desired frequencies over time.

In order to get the intensity, another algorithm was utilized, where the value of each sound frame is squared. Afterward, a Gaussian analysis window is used to convolve the result (*Praat - Sound: To Intensity...*, n.d.). This process returns an "intensity object", which contains the array of intensity values over time.

The two functions used to extract pitch and intensity can be configured using input parameters. For this project, the time between measurements (the "time step") was set to 0.01s for both functions, while the other parameters were left at their default values.

4

Results

Several different tests have been performed to collect the results. In this section, the results and significant figures will be presented. The findings from the tests concerning the amount of data are what was being used when training the networks to predict categories.

4.1 Amount of data

Several tests have been run to explore the impact of the amount of data on the result of the four different languages: English, Spanish, German, and French. The results can be found in Figure 4.1, which shows an increase in accuracy in connection with an increase in the number of files used. The same tests have been executed after adding Swedish as well. Those results can be seen in Figure 4.2.

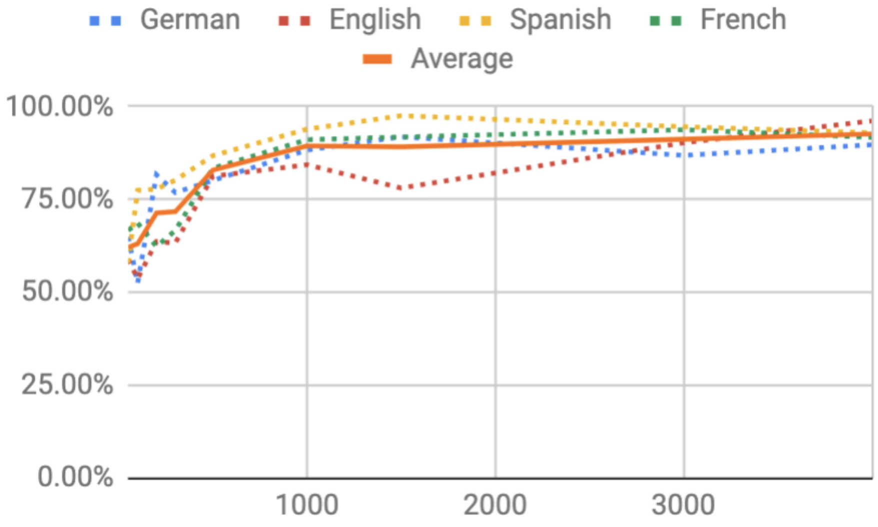


Figure 4.1: A graph showing how the accuracy increases with an increase in the amount of data used. A steady rise is visible up to 1000 files, thereafter the increase is significantly smaller. The tests are run for German, English, Spanish and French which all show the same trend.

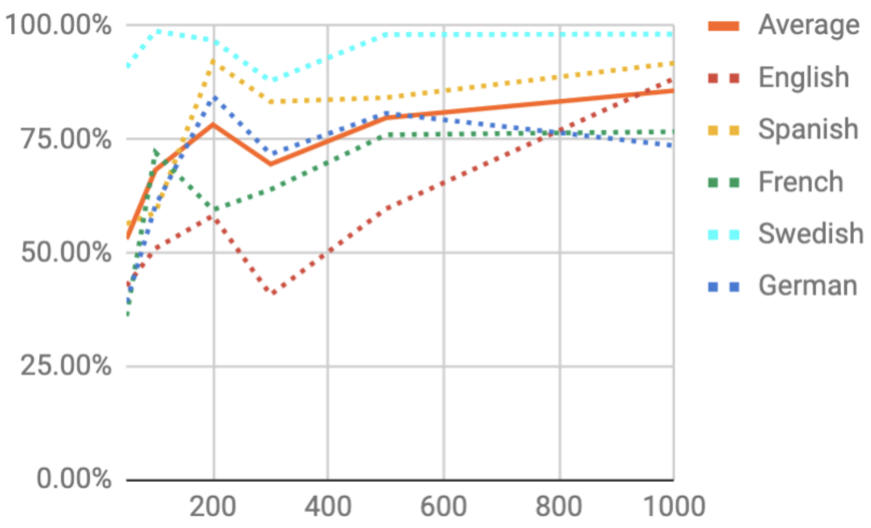


Figure 4.2: A graph showing how the accuracy increases with an increase in the amount of data used. The tests are run for Swedish, German, English, Spanish and French. Swedish can clearly be distinguished from the other languages with a much higher accuracy throughout the whole graph.

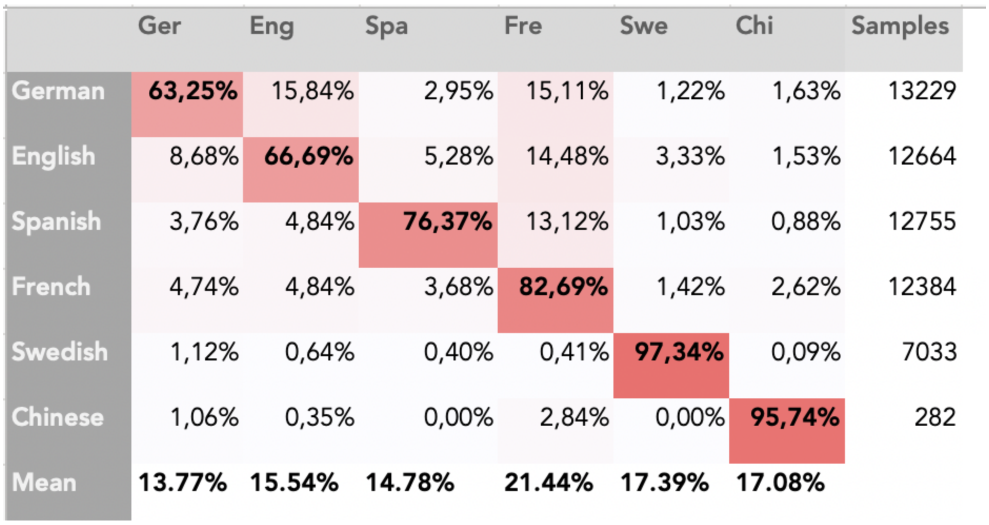
4.2 Language Categorization

The languages used when training the models were Swedish, German, French, Chinese, Spanish, and English. The Swedish audio data was only collected from the Riksdag while data in the other languages came from the audiobooks from LibriVox. In the following tables, the most predicted group for each actual group is marked with bold text. The different results produced are presented below.

4.2.1 LFLB-LSTM model

A network was trained on six languages and tested with all ten languages for the LFLB-LSTM model. A shown in table 4.1. Then, because of the deviation of Swedish and Chinese from the other languages, another network was trained on the remaining four languages and again tested with all ten.

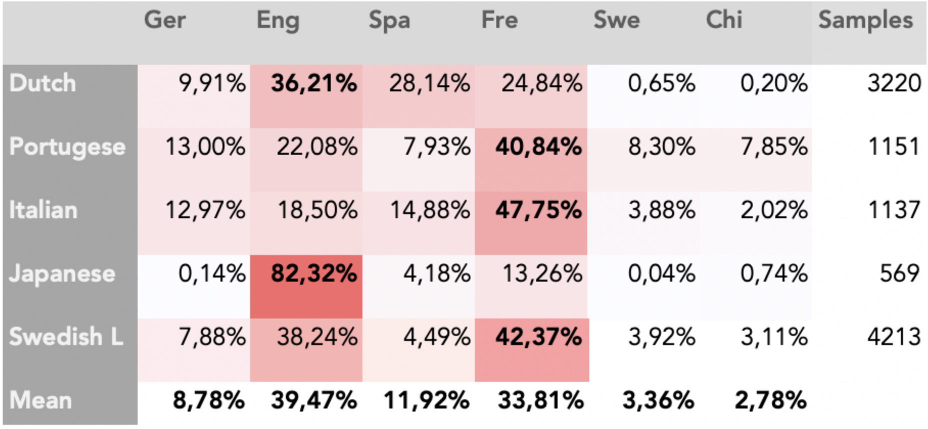
Table 4.1: The confusion matrix of the LFLB-LSTM model trained and tested on six different languages: German, English, Spanish, French, Swedish, and Chinese. In total, 2538 16-second samples (11 hours and 17 minutes of audio) from each language were used during training. The model showed the most confidence when predicting Swedish and Chinese, both with an accuracy above 90 %. The model has an average accuracy of 80.35 %.



	Ger	Eng	Spa	Fre	Swe	Chi	Samples
German	**63,25%**	15,84%	2,95%	15,11%	1,22%	1,63%	13229
English	8,68%	**66,69%**	5,28%	14,48%	3,33%	1,53%	12664
Spanish	3,76%	4,84%	**76,37%**	13,12%	1,03%	0,88%	12755
French	4,74%	4,84%	3,68%	**82,69%**	1,42%	2,62%	12384
Swedish	1,12%	0,64%	0,40%	0,41%	**97,34%**	0,09%	7033
Chinese	1,06%	0,35%	0,00%	2,84%	0,00%	**95,74%**	282
Mean	**13.77%**	**15.54%**	**14.78%**	**21.44%**	**17.39%**	**17.08%**	

Table 4.1 shows the confusion matrix for the LFLB-LSTM model trained with six different languages: German, English, Spanish, French, Swedish, and Chinese. The model has an average accuracy of 80.35 %. The model is most confident when predicting Swedish, where the recordings come from interpellations of the Riksdag. The model is also confident when predicting Chinese. Both Swedish and Chinese have an average accuracy above 90%.

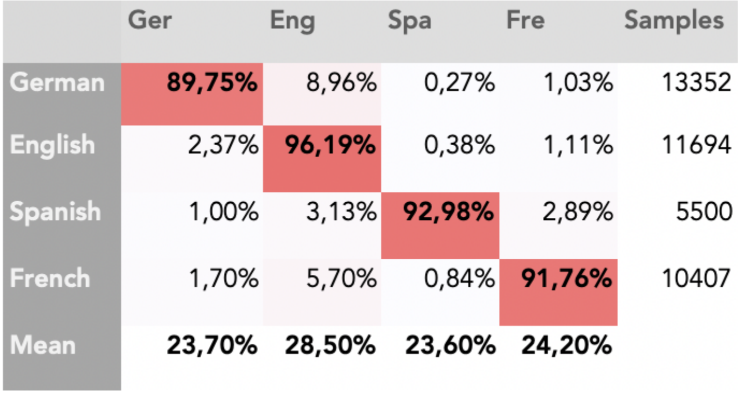
Table 4.2: The confusion matrix of the LFLB-LSTM model trained with six different languages and tested with several languages not used during training. The Swedish used for testing the network comes from audiobooks from LibriVox. The model is generally somewhat uncertain in the predictions and tends to predict English in 39.47 % of the times.



	Ger	Eng	Spa	Fre	Swe	Chi	Samples
Dutch	9,91%	**36,21%**	28,14%	24,84%	0,65%	0,20%	3220
Portugese	13,00%	22,08%	7,93%	**40,84%**	8,30%	7,85%	1151
Italian	12,97%	18,50%	14,88%	**47,75%**	3,88%	2,02%	1137
Japanese	0,14%	**82,32%**	4,18%	13,26%	0,04%	0,74%	569
Swedish L	7,88%	38,24%	4,49%	**42,37%**	3,92%	3,11%	4213
Mean	**8,78%**	**39,47%**	**11,92%**	**33,81%**	**3,36%**	**2,78%**	

Table 4.2 show the results of what the same neural network predicted when presented with other languages which it was not trained to recognize, namely Dutch, Portuguese, Italian, and Japanese. For these languages, the network could still only choose between the original six languages. These results were collected in order to see which languages it would pick and to try to understand what the network recognizes in the different categories. The model is generally somewhat uncertain in the predictions and tends to predict English in 39.47 % of the times.

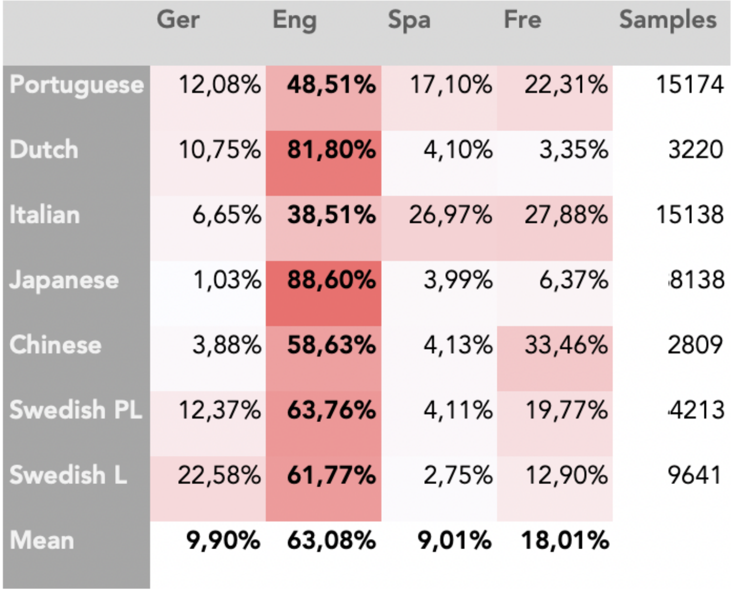
Table 4.3: The confusion matrix of the LFLB-LSTM model trained with four different languages: German, English, Spanish, and French. In total, 49,500 16-second samples (220 hours of audio) from each language were used during training. The model generates high accuracies for all languages, with an average accuracy of 92.67 %.



	Ger	Eng	Spa	Fre	Samples
German	**89,75%**	8,96%	0,27%	1,03%	13352
English	2,37%	**96,19%**	0,38%	1,11%	11694
Spanish	1,00%	3,13%	**92,98%**	2,89%	5500
French	1,70%	5,70%	0,84%	**91,76%**	10407
Mean	**23,70%**	**28,50%**	**23,60%**	**24,20%**	

The results of a test with the four different languages: German, English, Spanish, and French are shown in table 4.3. The model has high results for all languages and an average accuracy of 92.67 %. In total, 49,500 16-second samples (220 hours of audio) from each language were used during training.

Table 4.4: The confusion matrix of the LFLB-LSTM model trained with four different languages and evaluated by inserting seven languages that had not been used for training. The first row of Swedish is parliamentary interpellations from the Swedish Riksdag and the second row of Swedish is audiobooks from LibriVox. The model is generally uncertain in the predictions and predicts English 63.08 % of the times. Some tendencies to language relationships between Germanic and Romance languages can also be seen.



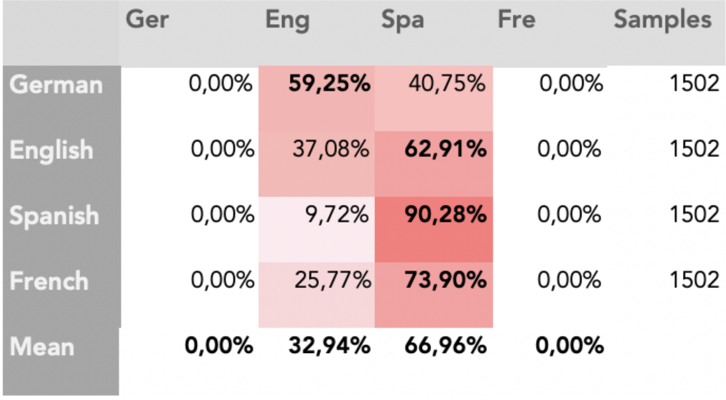
	Ger	Eng	Spa	Fre	Samples
Portuguese	12,08%	**48,51%**	17,10%	22,31%	15174
Dutch	10,75%	**81,80%**	4,10%	3,35%	3220
Italian	6,65%	**38,51%**	26,97%	27,88%	15138
Japanese	1,03%	**88,60%**	3,99%	6,37%	8138
Chinese	3,88%	**58,63%**	4,13%	33,46%	2809
Swedish PL	12,37%	**63,76%**	4,11%	19,77%	4213
Swedish L	22,58%	**61,77%**	2,75%	12,90%	9641
Mean	**9,90%**	**63,08%**	**9,01%**	**18,01%**	

Table 4.4 shows the results of the same neural network presented with languages which it was not trained to recognize, namely Dutch, Portuguese, Swedish from the Riksdag, Swedish from the LibriVox, Italian, Chinese and Japanese. For these languages, the network could still only choose between the original four languages. The results were collected in order to see which languages it would pick and to understand what the network recognizes between the different categories. The model is generally uncertain in the predictions and predicts English 63.08 % of the times. Some tendencies to language relationships between Germanic and Romance languages can also be seen by looking at Dutch or Swedish, that predicts English or German most often.

4.2.2 FFNN model

Solely one test was run for the FFNN model, due to the poor performance of the model it was not explored further.

Table 4.5: The confusion matrix of the FFNN model trained with four different languages: German, English, Spanish, and French. In total, 13 518 16-second samples (60 hours and 5 minutes of audio) from each language were used during training. The model solely predicts only two languages and reaches an average accuracy of 31.84 %.



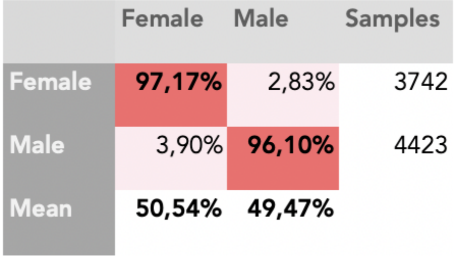
	Ger	Eng	Spa	Fre	Samples
German	0,00%	**59,25%**	40,75%	0,00%	1502
English	0,00%	37,08%	**62,91%**	0,00%	1502
Spanish	0,00%	9,72%	**90,28%**	0,00%	1502
French	0,00%	25,77%	**73,90%**	0,00%	1502
Mean	**0,00%**	**32,94%**	**66,96%**	**0,00%**	

The confusion matrix for the FFNN model trained with 1502 samples of German, English, Spanish, and French are shown in table 4.5. The average accuracy is 31.84%. As shown in table 4.5, the model only guesses between Spanish and English. Spanish is guessed 66.96% of the times and English 32.94% of the times.

4.3 Sex categorization

The sex models are trained solely with Swedish data gathered from the Riksdag since the other data source does not have a label for the sex of the speaker. The categories are male and female. The results are shown in table 4.6 and table 4.7.

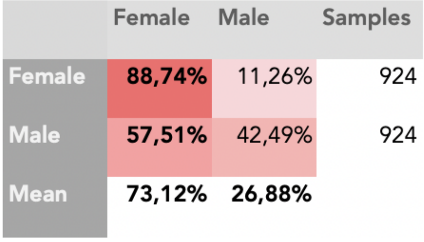
Table 4.6: The confusion matrix of the LFLB-LSTM model trained with audio files from male and female speakers. In total, 33 678 16-second samples (218 hours and 34 minutes of audio) from each language were used during training. The model shows extremely high accuracy when separating the two sexes.



	Female	Male	Samples
Female	**97,17%**	2,83%	3742
Male	3,90%	**96,10%**	4423
Mean	**50,54%**	**49,47%**	

Table 4.6 shows that the LFLB-LSTM network can clearly distinguish between sexes, with an average accuracy of 96.63% with a total of 9206 test samples. The network shows no tendencies to predict one sex more frequently than the other.

Table 4.7: The confusion matrix of the FFNN model trained audio files with an equal amount of male and female speakers. In total, 8316 16-second samples (36 hours and 58 minutes of audio) from each sex were used during training. The model can distinguish between male and female speakers but shows significantly higher probability to predict female speakers.



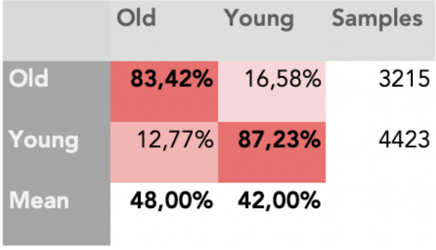
	Female	Male	Samples
Female	**88,74%**	11,26%	924
Male	**57,51%**	42,49%	924
Mean	**73,12%**	**26,88%**	

Table 4.7 shows that the FFNN model can distinguish between sexes, with an accuracy of 65.60% with a total of 1848 samples, 924 of each sex. The network shows tendencies to predict female speakers more often than male speakers.

4.4 Age Categorization

The age models are trained solely with Swedish data gathered from the Riksdag since the other data source does not have a label for the birth year. Two age groups were chosen to be used in the classification: a younger one, where the population is born after 1975 and an older one, where everyone is born before 1955. The results are shown in table 4.8 and table 4.9.

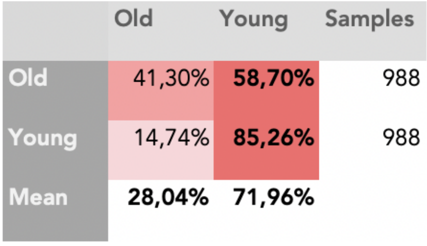
Table 4.8: The confusion matrix of the LFLB-LSTM model trained with two different age spans: born before 1955 and born after 1975. When training 642 samples have been used for each age span. The model has a higher accuracy for the older age span, which correlates with a higher probability to predict an elderly speaker.



	Old	Young	Samples
Old	**83,42%**	16,58%	3215
Young	12,77%	**87,23%**	4423
Mean	**48,00%**	**42,00%**	

Table 4.8 shows that the LFLB-LSTM model tends to predict elderly persons more often and therefore, also gets a higher accuracy on elderly persons. The average accuracy is 85,32%.

Table 4.9: The confusion matrix of the FFNN model trained with two different age spans: speakers born before 1955 and speakers born after 1975. When training 1976 samples have been used. The model has a higher accuracy for the younger age span, which correlates with a higher probability to predict a young speaker. The model can distinguish between the age spans and has an average accuracy of 63.24%.



	Old	Young	Samples
Old	41,30%	**58,70%**	988
Young	14,74%	**85,26%**	988
Mean	**28,04%**	**71,96%**	

Table 4.9 shows that the FFNN model tends to predict younger speakers more often and therefore, also gets a higher accuracy on younger persons. The model can distinguish between the two age groups and has an average accuracy of 63.24%.

5 Discussion

The results of the implementation of the two different models will be discussed in this section. Further, the choice of data collection, method, and the tools used in the project will be addressed. After that follows a view of the ethical aspects and the possible impact the product could have on the society and ideas regarding future work within the subject.

5.1 Amount of data

The tests show that accuracy is correlated with the amount of data. A greater amount of data generates a higher accuracy. The best results for the LFLB-LSTM model with four languages was achieved when training on 4000 files per language, but it is also clear that the relationship is not linear (see figure 4.1). A thousand files gave an almost identical accuracy score, which seems to indicate that it would not be necessary to gather as much data if another language were to be included in a model. However, the nature of the language would also affect how much data were to be needed. The LFLB-LSTM model with six languages returned good results for Chinese although there were only 235 such files available. Presumably, this is due to the fact that it differs so much from the other languages, which were all European.

Another remarkable aspect of these tests is the fact that Swedish achieved such high accuracy after very little training. A model trained on German, English, Spanish, French and Swedish with just 50 files per language achieved an accuracy of 90.91% for Swedish, much higher than for any of the others (see figure 4.2). This result suggests that the Swedish dataset differs significantly from the others in some way, which will be discussed further in section 5.2.

5.2 Language categorization

There could be multiple reasons behind the less optimal result of the feature model. The dataset in this study was more substantial compared to the study from J.C de Bruin and J.A du Preez, but they had a greater insight into their dataset. It is vital that the data is preprocessed in a way that makes it suitable for feature selection. Intensity features were added, and the tone features were excluded, and this could potentially have made the subtle differences between the languages less visible.

J.C. de Bruin and J.A. du Preez stated in their paper when explaining their poor English results that a reason for the outcome could have been the lack of native speakers, which might confuse the model. There has been no way to guarantee that the speakers are native speakers in any of the audio data used when training the models in this project, which could pose as a significant source of error.

Not a number (NaN) values were present in the pitch and intensity array of the datasets as well at times. The audio sample was omitted when this occurred. If this was a case of data missing at random (MAR), which means that the data is missing due to one of its properties, this could potentially introduce bias to the dataset. This approach would cause it to struggle to classify other data than it has seen before. A solution to this is to investigate the data and impute the missing value with a suitable imputation method.

A further problem with the language categorization is that the audiobooks from LibriVox do not have a label for the sex or birth year of the speaker, which means there is no guarantee that the sexes and ages are balanced among the different languages. This scenario could result in a neural network relating the sex or age of a speaker to a particular language, instead of focusing on the language itself.

Another thing that is worth noting is that the Swedish audio files are collected from the Swedish Riksdag while the other languages are all recorded audiobooks by individual volunteers. The differences in audio quality, as well as the differences in the type of speech, could explain why Swedish is so easily distinguished compared to the other languages. Since all Swedish audio files were recorded in the same environment and probably with the same equipment, the training data may contain less variation in Swedish than in other languages which in turn makes the networks worse at recognizing Swedish outside of the Riksdag. The model tends to predict French or English when presented with Swedish audio files from LibriVox or other sources than the Swedish Riksdag, which is a further indicator of the importance of variation within the data set.

A method used for analyzing the LFLB-LSTM models was to test the model with languages it had never seen before, such as Portuguese, Italian, Dutch, and Japanese. The models expressed a greater uncertainty than otherwise, but the predicted results were somewhat expected considering the relationship between the languages. A reason to why Dutch was recognized as English could be since they are both Germanic stress-timed languages. German also fits this description, but since the model has earlier showed tendencies to predict English if it is uncertain, a language that is related to both English and German are likely to be predicted as English more often. The reason for this may stem from the fact that there were a higher number of unique speakers represented in the English dataset and therefore a higher degree of variance. This situation may have resulted in the network learning what the three other languages sounded like and then merely classifying everything else as English. This potential scenario would explain the high confidence when testing for Japanese since the language is very different from the others. It is neither stress-timed nor syllable-timed (Mok, 2009), and it does not share any relationships with the other languages in terms of language-families. More testing would have to be conducted in order to confirm such a hypothesis. For the romance languages Italian and Portuguese, the model ordinarily predicted the Romance languages but also showed rather high results for English, again possibly due to the reasons mentioned above.

5.3 Sex categorization

The LFLB-LSTM model and the Feature model indicate that categorizing based on sex is possible. An explanation to the higher performance of the LFLB-LSTM network compared to the FFNN network, apart from the more advanced architecture, is that the pitch and intensity sequences are left unmodified.

Another thing to have in mind is that the idea was to extract prosodic features and try to categorize different categories based on prosody. Since the pitch is equivalent to the fundamental frequency in the project, and males' and females' vocal cords often differ due to physical differences between the sexes (Eulenberg & Farhad, 2011), it is possible to believe that the ANN chose to separate the sexes based on their physical differences rather than the different intonation between men and women.

5.4 Age categorization

The LFLB-LSTM model and the FFNN model indicate that categorizing based on two age categories, separated by 20 years, is possible. The differentiation is only done on two age spans, speakers born before 1955 and speakers born after 1975. The reason for this is limited access to speakers of different ages from the Swedish Riksdag, which is used as training data for the age models. Speakers born between these years were omitted to create some distance between the groups so the neural network can recognize them more reliably.

In figure 5.1 the distribution of the birth years of the speakers of the Riksdag interpellations is shown, along with the years at which the groups were separated. As can be seen, most audio clips which were used have speakers born soon before 1955 or soon after 1975.

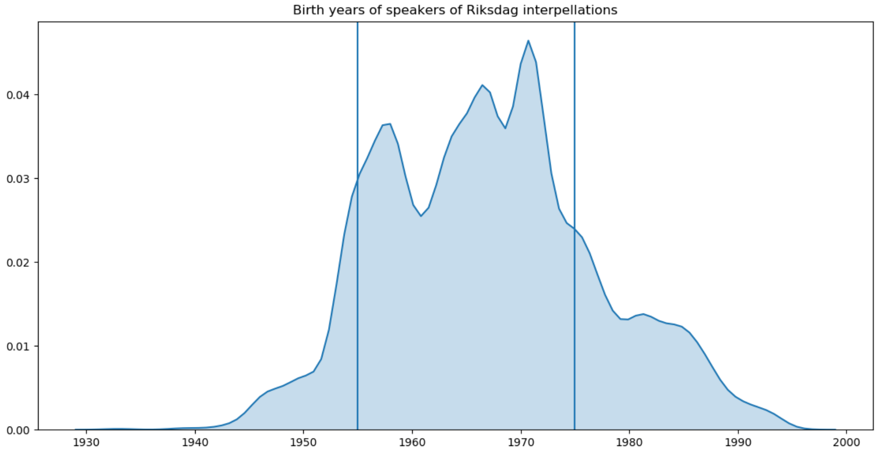


Figure 5.1: The distribution of the birth years of the speakers of Riksdag interpellations shown along with the years at which we separate the groups.

5.5 Datasets

The available datasets for this project were limited. The hope early on was to find one or more readily labeled datasets, which could be used out of the box to train a neural network. As no such satisfactory dataset was discovered, other methods were pursued, eventually leading to the use of the Swedish Riksdag and LibriVox as sources.

Gathering data from two such different sources as Riksdag speeches and audiobook recordings could pose a problem. Primarily due to the difference in audio quality and the difference in speaking style when reading an audiobook or discussing politics in front of a room full of people.

5.5.1 LibriVox

The LibriVox dataset, although very valuable as a large source of open data, still has its shortcomings. Since much of the information about the speakers is entirely unknown, it is difficult to know how the speakers vary in terms of age, gender, and dialects. If there were to be a very skewed sex balance that differs between languages, it is likely that the network would attribute male and female speakers to different languages. A reliable dataset should accurately reflect the population it is trying to represent.

Another problem with this dataset is that it contains a lot of different genres, among them music, language learning, and poetry. This fact might be very troublesome, but since these categories constitute such a small cut of the dataset, especially in the languages that concerned this project, and demanded much work to be removed, this was not prioritized. This flaw is something that definitely should be improved if one poses to do similar work. Further, since individual volunteers record the audiobooks with their equipment, the audio quality could be inferior.

5.5.2 The Swedish Riksdag

The API of the Swedish Riksdag was easily accessible and well structured. The recorded audio files also provided an advantage of being recorded by the same microphone and holding equally high audio quality.

The main issue with this dataset is the type of audio files being used. Only Riksdag speeches are used as training data since this could guarantee that only one person spoke at a time, which ensured that the language-specific prosodic features were not confused with multiple speakers. The data recordings are only available from 2011 though, which results in several unique speakers that are limited by the number of members of the Riksdag since 2011.

Another problem with the dataset is that after a question to the minister, there is a pause while they change places so that the minister can answer the question. These pauses are not very long but quite frequent and might pose a problem to the categorization partly because of the silence and partly because of the possible background noises. The style of speech does not vary very much in parliament, which also be should be taken into account.

5.6 Real world experiments

The models have not been thoroughly tested with data from other sources than LibriVox and the Swedish Riksdag due to a lack of time. Early spot-checks indicate that samples from radio shows, interviews, and recordings return worse results than returned by the test data. This fact is not unexpected since a neural network only learns to navigate in situations that it has encountered before. Since it has only been trained on audiobook recordings and parliamentary interpellations those are the circumstances under which the best results were achieved. More tests are needed to verify this.

During the exhibition, the model that was trained on 220 hours of audio in each of the four languages: English, Spanish, French, and German, was tested with several speakers reading a text in an optional language in 16 seconds. The model generally predicted English, no matter which language was being spoken, unless the speaker was a native speaker. Five unique native speakers tried the demo, one speaker from each language and the last with origins in South America. The model managed to categorize four out of these five speakers correctly, with the Argentinian being the exception.

5.7 Audio preprocessing

A method, by which to extract that data from an audio file and save it, was necessary, preferably one which could be automated to avoid time-consuming and repetitive work. After having tried out various audio software tools such as Audacity and Sonic visualizer, the audio engineering tool Praat was found to be useful for the purposes of this project (*Praat: doing Phonetics by Computer*, n.d.). One method which was explored involved taking the source audio file, passing it to Praat, and using the built-in features of that program to extract a "pitch object", and converting it to a "pitchtier object", which was then exported to a text file. After that, the contents of the text file were converted into an array through a python script using a parser, which was purpose-built for the way these text files were laid out.

This process required every audio file to be individually and manually passed through "Praat," which did not satisfy the desired autonomy of the process. Therefore another method was devised which made use of an existing python library known as Parselmouth, which aims to make the functions of the program Praat available in a python setting (Jadoul et al., 2018). This tool enabled the process to be automated entirely, by use of the functions "to_pitch" and "to_intensity" as explained in section 3.7.3. By taking out the correct data from the results of these functions, arrays of pitch and intensity of the audio over time can be created. These arrays were then ready to be used as input data for a neural network.

5.8 Recreation of prosody

The extraction of the prosody was done with the help of algorithms that could be set up using some different parameters. Here different parameters were not tested. Instead, the default values were used for most of them (except the time between measurements, 0.01 s). Using different values may have led to a better result.

For the prosody extraction in this project, pitch and intensity were extracted. There may be other exciting metrics that could be used alongside those which may be able to improve the results further. There are also additional aspects of prosody which would have been interesting to utilize, such as stress and length of vowels. They were not used as no reliable way of automatically retrieving those aspects on large amounts of data were found.

5.9 Poor results of FFNN model

The tests with the FFNN model rendered worse results than with the LFLB-LSTM model, in some cases even accuracies beneath 50%. A possible reason for this poor result is that the selected features did not vary between groups as much as expected. To view an example, see figure 5.2, which shows that there are no dramatic differences between the groups along with the metric of the average gradient of positive slopes.

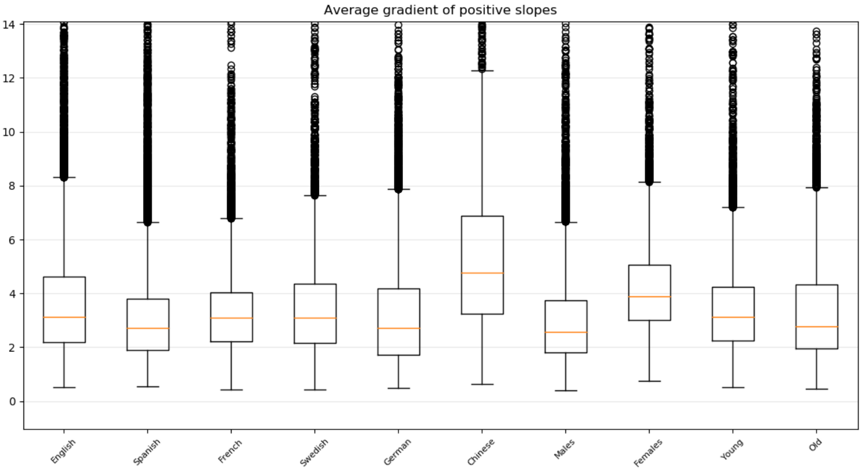


Figure 5.2: Boxplots of the distribution of the average gradient of positive slopes (Hz/s) for each group.

The groups of people present here are speakers of the languages Swedish, English, German, Spanish, French, and Chinese, along with men, women, young people (born after 1975) and old people (born before 1955) from the Riksdag. Boxplots of the distributions of more features for these groups of people can be found in appendix A. Another possibility is that these features were not appropriately extracted and that some of them, therefore, had invalid values which negatively affected the result.

5.10 Social and Ethical Aspects

A lot of ethical aspects have required our attention during this project. Even though all the data that has been used is publicly available, aspects regarding what is ethically correct to save and use when training our networks have been a subject of discussions. For example, when saving the personal information from the Swedish Riksdag, the political opinion of each speaker was accessible, as well as a full record of education and the employment of the respective parents. We did not feel comfortable saving this rather sensitive information or using when predicting our categories since it could easily be misused for other purposes.

A concern with the final program could be if it were used in a manner that would treat people differently depending on the language they spoke or what sex they are. The issue with categorizing any trait is that it could be used in a discriminating way, which is something that further segregates society. Can a machine with the capability of treating people differently depending on their background be considered ethical? Perhaps insurance companies could use this information to determine the rates based on a person's speech? It might even be possible for authoritarian governments to locate and harass ethnic minorities by monitoring how people speak when they are using their phones.

5.11 Future work

An interesting subject for the future would be to further look more into detail of the categorization based on prosody, and work with the essential details of prosody together with carefully selected data. By looking at shorter audio sequences, one could try to manipulate the sound and see how the neural network adapts and categorizes the new corrupted files. These tests could increase one's knowledge of how the network is processing the audio files, which later could improve one's ability to create more powerful models as well as learning about significant features that humans potentially overlook.

Other categories that were discussed during this project were accents, a natural category to branch to after looking at languages. Emotional state and intent was also brought up as an interesting subject to look into, which could be used by smart agents to improve the user experience of the interaction.

To categorize questions vs. statements was also discussed as potential future work. One of the issues with categorizing questions vs. statements is that the prosody commonly varies from languages to languages and accents to accents, such as how a question or statement is pronounced (Nordquist, 2018). In English, it is common that the pitch decreases at the end of a statement and increases at the end of a question. Australian English, however, uses high rising terminals, which means that the pitch often rises at the end of a statement. This style of speaking is also typical in California, where it has come to be known as valley girl speech.
This problem could potentially be solved by adding a pipeline of models that first categories the language and regional accent, followed by a model that classifies the sentence as a question or statement. This obstacle would also be an exciting project to look into one day, where many different models of classification could be added, returning some status of a person based on their prosody.

6 Conclusion

Based on the research and experiments done in this project it is possible for the LFLB-LSTM model to distinguish

- Swedish, English, Spanish, French, German, and Chinese from one another.
- the sexes male and female from one another.
- the age groups *born after 1975* and *born before 1955* from one another.

Based on the research and experiments done in this project it is possible for the FFNN model to distinguish

- the sexes male and female from one another.
- the age groups *born after 1975* and *born before 1955* from one another.

The network draws these conclusions based solely on the pitch and the intensity of the audio files.

The language model is introduced to a few new languages, and it returns a relatively expected result. Germanic languages are a vast majority of the times classified as Germanic languages English and German, while Romance languages are recognized as the Romance languages French or Spanish. If the model is uncertain, such as the case with Japanese, it classifies the language as English which could be due to the large number unique speakers in the English data set that was used when training the LFLB-LSTM model. These results support the idea that the prosody could be taken into account. Prosody as a whole might make these predictions in the language model, but this can not be guaranteed.

The feature model performed significantly worse than the LFLB-LSTM model, but potentially not just because of its architecture, but rather a lack of understanding of what it was in the data that set the groups apart from one another, an important criterion when creating a suitable model for a dataset.

Since the data from the Swedish Riksdag generated a network that was not able to categorize other Swedish speech, it highlights the importance to get a sufficient variance in the data within groups to avoid mistakes of selecting unrelated properties from the data. The neural network will less frequently categorize based on unrelated properties when the audio has a higher variance, in terms of audio quality, type of reading, emotional state, and other similar characteristics.

7

Bibliographic notes

Similar work

- (Pi school, 2017) - A school that works with innovation and artificial intelligence. They worked with speech recognition with six languages and achieved quite satisfying results.
- (Vicsi & Szaszák, 2010) - Klára Vicsi and György Szaszák, from the Laboratory of Speech Acoustics, in Budapest University of Technology and Economics, did a study on how to use prosody to improve speech categorization.
- (de Bruin & du Preez, 1993) - A South African study conducted in 1993 on pitch differences between some languages. Inspiration was drawn from this study about what linguistic features to extract from the pitch curves etc.
- (Graves et al., 2013) - Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton from the computer science department in the University of Toronto trained a recurrent neural network model for speech recognition back in 2013 and reported their results.
- (Kamble, 2016) - Bhushan C. Kamble gives a broad review over the common practises in speech recognition with neural networks and other mathematical models.
- (Burrows & Niranjan, 1994) - T. L. Burrows M. Niranjan from the computer engineering department of Cambridge University investigates the use of recurrent neural networks for classification.
- (Zhao et al., 2019) - A paper which introduces the concept of LFLBs and utilises them to classify speech emotion. The paper which our LFLB-LSTM model is based on.

Speech

- (Manell, 2008a, 2008b) - The Department of Linguistics, a part of the Faculty of Human Sciences at Macquarie University in Australia, are doing research in language acquisition and disorders of language; language, speech, and hearing, as well as languages in society.
- (Vajda, 2001) - An overview over prosody, Linguistics 201, held by professor Edward Vajda at Western Washington University.
- (Oxenham, 2012) - Andrew J. Oxenham from the Department of Psychology, University of Minnesota, wrote a paper on pitch perception in the Journal of Neuroscience.
- (Gibbon, 2017) - A paper by Dafydd Gibbon from the University of Bielefeld in Germany. Gibbon is talking about rhythm and the melodies of speech,

based on a previous course called Contemporary Phonetics and Phonology at Tongji University, Shanghai, China.
- (Conlen, 2016) - Madeline M. Conlen from Wayne State University made a linguistic comparison on stress-timed and syllable-timed languages.
- (Frankfurt International School, n.d.) - Frankfurt International School is a language school, where they on this page provide the differences between English and Swedish.
- (School, n.d.) - Frankfurt International School is a language school, where they on this page provide the differences between English and Portuguese
- (British Council, n.d.) - The British Council is an organization that works internationally with culture and languages. In this article, they discuss stress-timed languages.
- (Mok, 2009) - Peggy Pik Ki Mok from the Chinese University of Hong Kong discusses the syllable-timing of Cantonese and Beijing Mandarin, as well as the rhythm of other languages.
- (Collins & Mees, 1984) - Beverly Collins and Inger Mees discuss stress-timed and syllable-timed languages in their book *The Sounds of English and Dutch.*
- (Hartmann, 1997) - Signals, Sound, and Sensation, a book written by William Morris Hartmann where he talks about psychoacoustics in pitch perception.
- (Li & Jain, 2009) - Li, Stan Z. and Jain talks about what the fundamental frequency is and how it is linked together with pitch.
- (Wolfe, 2019) - Joe Wolfe from the University of New South Wales talks about the physical properties of prosody.
- (Eulenberg & Farhad, 2011) - A resource from the University of Michigan with information about the human voice.
- (Butler et al., 2013) - Research on how the voice changes over the years, used to support the claims regarding how the network can distinguish age differences.
- (Nordquist, 2018) - An article about speech patterns such as uptalking.

Data sources
- (*Dokumentation - Riksdagens öppna data*, n.d.) - Documentation for the open data made available by the Swedish Riksdag which includes an API used to get Swedish mp3 files.
- (*LibriVox*, n.d.) - A website hosting audio files of readings of audiobooks in various languages. This website was a source of mp3 files in languages other than Swedish.

Audio preprocessing
- (*Praat: doing Phonetics by Computer*, n.d.) - A piece of audio engineering software, capable of extracting a pitch or an intensity curve from an audio file. Early in the project, this software tool was utilized in order to convert from audio files to pitch curves, which are to be used with the neural network.
- (*Praat - Sound: To Pitch (ac)...*, n.d.) - A function in Praat to extract the fundamental frequency from an audio file. Used to obtain pitch arrays.
- (Boersma, 1993) - Presents a method for accurate short-term analysis of the fundamental frequency. Used in the Praat function "Sound: To Pitch (ac)".

- (*Praat - Sound: To Intensity...*, n.d.) - A function in Praat to extract the intensity from an audio file. Used to obtain intensity arrays.
- (Jadoul et al., 2018) - Presents the python library parselmouth, which aims to implement the functions of Praat in python.

Neural networks

- (Mehlig, 2019) - Lecture notes from the course FFR135 by Bernard Mehlig at Chalmers University of Technology.
- (Smith, 2018) - A research laboratory technical report that presents a disciplined approach to Neural Network hyperparameters.
- (Srivastava et al., 2014) - A paper that presents dropout as a technique to prevent overfitting in Deep Learning.
- (Albawi et al., 2017) - A conference paper presented at 2017 International Conference on Engineering and Technology (ICET) on the topic of understanding convolutional neural networks.
- (Yamashita et al., 2018) - An article published in "Insights into Imaging" concerning convolutional neural networks and their application in radiology.
- (Kiranyaz et al., 2015) - A paper were a new method for electrocardiogram (ECG) classification that uses one dimensional convolutional networks is proposed.
- (Abdeljaber et al., 2017) - A method discovering structural damage on constructions that makes use of one dimensional convolutional networks.
- (Hochreiter & Schmidhuber, 1997) - The paper which first introduced the RNN architecture Long Short-Term Memory networks
- (Lipton, 2016) - An article discussing why it is so difficult to tell what is going on within an artificial neural network.
- (Albawi et al., 2017) - A paper written as an introduction to the CNN architecture.
- (Yamashita et al., 2018) - An introduction to the CNN architecture and its usecases in radiology.
- (Ioffe & Szegedy, 2015) - The paper that introduced the concept of batch normalization.
- (Zyner et al., 2017) - An implementation of the LSTM architecture where it is used to predict the behaviour of drivers in real-time.
- (Graves et al., 2013) - An implementation of the recurrent architecture LSTM which is used to determine the language of speech.
- (Graves et al., 2009) - A RNN used to predict the movements of numbers on a screen.
- (Li & Wu, 2014) - A speech recognition system based on the LSTM architecture.
- (Sherstinsky, 2018) - An introduction to recurrent neural networks.
- (Mary & Yegnanarayana, 2008) - A report doing similar work, extracting prosodic features to distinguish between languages. Used as inspiration when testing different network architectures for the FFNN.
- (Leena et al., 2005) - A similar study of speech recognition based on prosodic

features, that was used as inspiration when testing different network architectures for the FFNN.
- (Mary & Yegnanarayana, 2008) - A study that performed similar work by determining languages based on prosody, used as inspiration when testing different network architectures for the FFNN.

Tools

- (*The Python Programming Language*, n.d.) - A high-level programming language used for all the code in the project.
- (Chollet et al., 2015) - Documentation for keras, a deep learning python library used for implementation of neural networks.
- (*Parselmouth – Praat in Python, the Pythonic way*, n.d.) - Documentation for the python library parselmouth, describing its functions. Used for our audio preprocessing.

References

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, 154 - 170. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0022460X16306204` doi: https://doi.org/10.1016/j.jsv.2016.10.043

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, Aug). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (icet)* (p. 1-6).

Aphex34. (2015). *Typical cnn.png.* Wikipedia. Retrieved from `https://en.wikipedia.org/wiki/File:Typical_cnn.png`

Boersma, P. (1993). *Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound.* Retrieved from `http://www.fon.hum.uva.nl/paul/papers/Proceedings_1993.pdf`

British Council. (n.d.). *Stress timed.* Retrieved from `https://www.teachingenglish.org.uk/article/stress-timed`

Burrows, T. L., & Niranjan, M. (1994). *The use of recurrent neural networks for classification.* Retrieved from `ftp://mi.eng.cam.ac.uk/pub/reports/auto-pdf/burrows_nnsp94.pdf`

Butler, A., Lind, V., & Weelden, K. V. (2013). Research on the aging voice: Strategies and techniques for healthy choral singing. *The Phenomenon of Singing*, *3*(0). Retrieved from `https://journals.library.mun.ca/ojs/index.php/singing/article/view/627`

Chollet, F., et al. (2015). *Keras.* `https://keras.io`.

Collins, B., & Mees, I. (1984). *The sounds of english and dutch.* E.J Brill / Leiden University Press.

Conlen, M. M. (2016). *A linguistic comparison: Stress-timed and syllable-timed languages and their impact on second language acquisition.* Retrieved from `https://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1025&context=honorstheses`

de Bruin, J. C., & du Preez, J. A. (1993, Aug). Automatic language recognition based on discriminating features in pitch contours. In *1993 ieee south african symposium on communications and signal processing* (p. 133-138). doi: 10.1109/COMSIG.1993.365857

Dokumentation - riksdagens öppna data. (n.d.). Retrieved from `https://data.riksdagen.se/dokumentation/`

Eulenberg, J., & Farhad, A. (2011). *Fundamental frequency and the glottal* *pulse.* Retrieved from https://msu.edu/course/asc/232/study_guides/F0_and_Glottal_Pulse_Period.html

Frankfurt International School. (n.d.). *The differences between english and swedish.* Retrieved from http://esl.fis.edu/grammar/langdiff/swedish.htm

Gibbon, D. (2017). *Prosody: Rhythms and melodies of speech.* Retrieved from https://arxiv.org/pdf/1704.02565.pdf

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009, May). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855-868. doi: 10.1109/TPAMI.2008.137

Graves, A., Mohamed, A., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 ieee international conference on acoustics, speech and signal processing* (p. 6645-6649).

Graves, A., rahman Mohamed, A., & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks.* Retrieved from http://www.cs.toronto.edu/~hinton/absps/DRNN_speech.pdf

Hartmann, W. M. (1997). *Signals, sound, and sensation.* ISBN 978-0-387-23472-4.

Hochreiter, S., & Schmidhuber, J. (1997, November). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780. Retrieved from http://dx.doi.org/10.1162/neco.1997.9.8.1735 doi: 10.1162/neco.1997.9.8.1735

Ioffe, S., & Szegedy, C. (2015, Feb). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv e-prints*, arXiv:1502.03167.

Jadoul, Y., Thompson, B., & de Boer, B. (2018). Introducing parselmouth: A python interface to praat. *Journal of Phonetics*, *71*, 1 - 15. Retrieved from http://www.sciencedirect.com/science/article/pii/S0095447017301389 doi: https://doi.org/10.1016/j.wocn.2018.07.001

Kamble, B. C. (2016). *Speech recognition using artificial neural network – a review.* Retrieved from http://iieng.org/images/proceedings_pdf/U01160026.pdf

Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015, Aug). Convolutional neural networks for patient-specific ecg classification. In *2015 37th annual international conference of the ieee engineering in medicine and biology society (embc)* (p. 2608-2611). doi: 10.1109/EMBC.2015.7318926

Konstantin-Klemens, L. (2018). *Natural language processing in artificial neural networks* (Unpublished doctoral dissertation). Lund University.

Leena, M., Srinivasa Rao, K., & Yegnanarayana, B. (2005, Jan). Neural network classifiers for language identification using phonotactic and prosodic features. In *Proceedings of 2005 international conference on intelligent sensing and information processing, 2005.* (p. 404-408). doi: 10.1109/ICISIP.2005.1529486

Li, S. Z., & Jain. (2009). *Fundamental frequency, pitch, f0.* Retrieved from https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-73003-5_775

Li, X., & Wu, X. (2014, Oct). Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition. *arXiv e-prints*, arXiv:1410.4281.

Librivox. (n.d.). Retrieved from https://librivox.org/

Lipton, Z. C. (2016, Jun). The Mythos of Model Interpretability. *arXiv e-prints*, arXiv:1606.03490.

Manell, R. (2008a). *Introduction to prosody theories and models.* Retrieved from https://www.mq.edu.au/about/about-the-university/faculties-and-departments/faculty-of-human-sciences/departments-and-centres/department-of-linguistics/our-research/phonetics-and-phonology/speech/phonetics-and-phonology/intonation-prosody

Manell, R. (2008b). *Speech waveforms.* Retrieved from http://clas.mq.edu.au/speech/acoustics/waveforms/speech_waveforms.html

Mary, L., & Yegnanarayana, B. (2008). Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication*, *50*(10), 782 - 796. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167639308000587 doi: https://doi.org/10.1016/j.specom.2008.04.010

Mary, L., & Yegnanarayana, B. (2008, Jan). Prosodic features for language identification. In *2008 international conference on signal processing, communications and networking* (p. 57-62). doi: 10.1109/ICSCN.2008.4447161

Mehlig, B. (2019, Jan). Artificial Neural Networks. *arXiv e-prints*, arXiv:1901.05639.

Mok, P. P. K. (2009). *On the syllable-timing of cantonese and beijing mandarin.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.584.7076&rep=rep1&type=pdf

Nordquist, R. (2018). *Speech patterns: Uptalking.* Retrieved from https://www.thoughtco.com/uptalk-high-rising-terminal-1692574

Oxenham, A. J. (2012). *Pitch perception.* Retrieved from http://www.jneurosci.org/content/jneuro/32/39/13335.full.pdf

Parselmouth – praat in python, the pythonic way. (n.d.). Retrieved from https://parselmouth.readthedocs.io/

Pi school. (2017). *Spoken language identification.* Retrieved from https://picampus-school.com/spoken-language-identification/

Praat: doing phonetics by computer. (n.d.). Retrieved from http://www.fon.hum.uva.nl/praat/

Praat - sound: To intensity... (n.d.). Retrieved from http://www.fon.hum.uva.nl/praat/manual/Sound__To_Intensity___.html

Praat - sound: To pitch (ac)... (n.d.). Retrieved from http://www.fon.hum.uva.nl/praat/manual/Sound__To_Pitch__ac____.html

The python programming language. (n.d.). Retrieved from https://www.python.org/

School, F. I. (n.d.). *The differences between english and portuguese.* Retrieved from http://esl.fis.edu/grammar/langdiff/portuguese.htm

Sherstinsky, A. (2018, Aug). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *arXiv e-prints*, arXiv:1808.03314.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *CoRR*, *abs/1803.09820*. Retrieved from http://arxiv.org/abs/1803.09820

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Vaissière, J. (1983). Language-independent prosodic features. In A. Cutler & D. R. Ladd (Eds.), *Prosody: Models and measurements* (pp. 53–66). Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-69103-4_5 doi: 10.1007/978-3-642-69103-4_5

Vajda, E. (2001). *Prosody (suprasegmental features).* Retrieved from http://pandora.cii.wwu.edu/vajda/ling201/test2materials/prosody.htm

Vicsi, K., & Szaszák, G. (2010). Using prosody to improve automatic speech recognition. *Speech Communication*, *52*(5), 413 - 426. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167639310000129 doi: https://doi.org/10.1016/j.specom.2010.01.003

Wolfe, J. (2019). *Frequency and pitch.* Retrieved from http://www.animations.physics.unsw.edu.au/jw/frequency-pitch-sound.htm

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018, Jun 22). Convolutional neural networks: an overview and application in radiology. *Insights Imaging*, *9*(4), 611-629. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/29934920 (29934920[pmid]) doi: 10.1007/s13244-018-0639-9

Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical Signal Processing and Control*, *47*, 312 - 323. Retrieved from http://www.sciencedirect.com/science/article/pii/S1746809418302337 doi: https://doi.org/10.1016/j.bspc.2018.08.035

Zyner, A., Worrall, S., Ward, J., & Nebot, E. (2017, June). Long short term memory for driver intent prediction. In *2017 ieee intelligent vehicles symposium (iv)* (p. 1484-1489). doi: 10.1109/IVS.2017.7995919

A

Boxplots of prosodic features

Presented in this appendix are boxplots of the distributions gathered for each group for the 14 calculated prosodic features. For each group, 1000 values were used for every feature, except for Chinese, which had only 235 values for every feature.

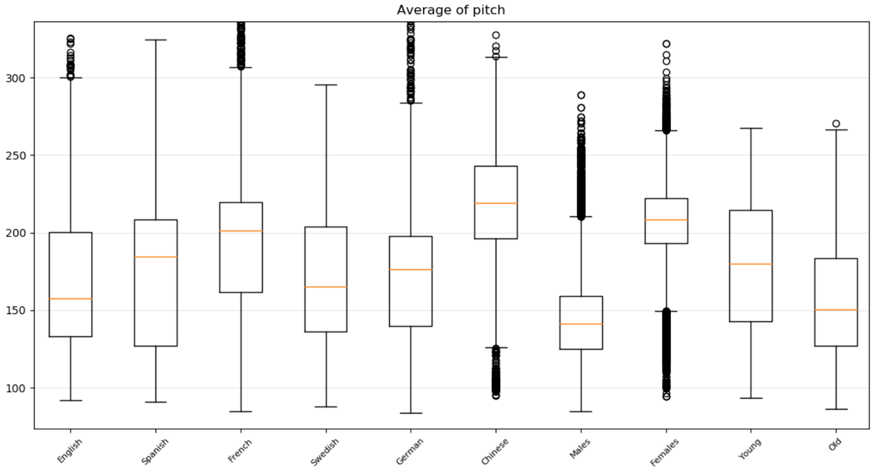


Figure A.1: Boxplots of the distribution of the average pitch (Hz) for each group.

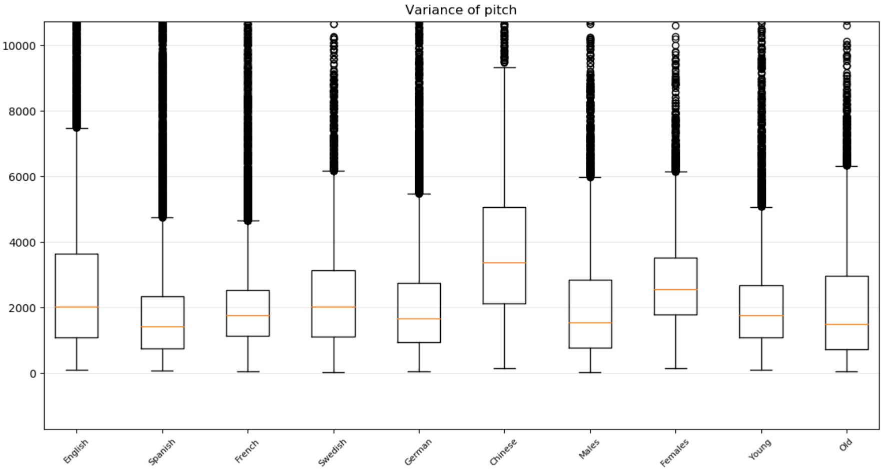


Figure A.2: Boxplots of the distribution of the pitch variance (Hz^2) for each group.

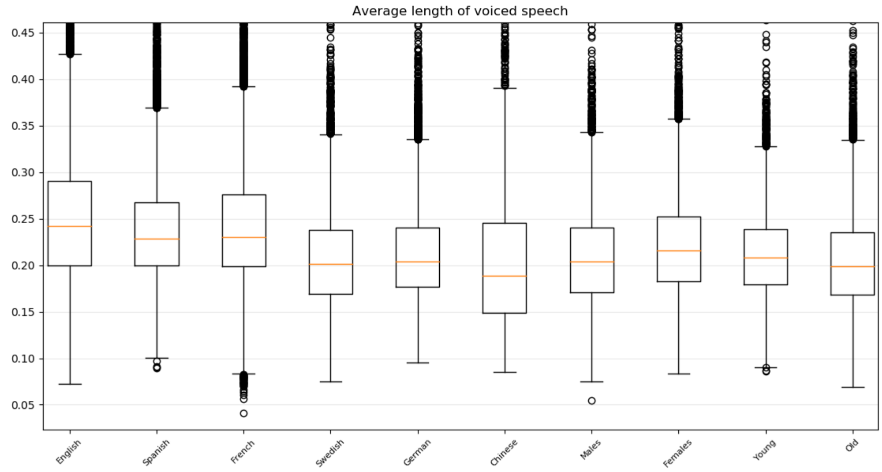


Figure A.3: Boxplots of the distribution of the average length of voiced speech (s) for each group.

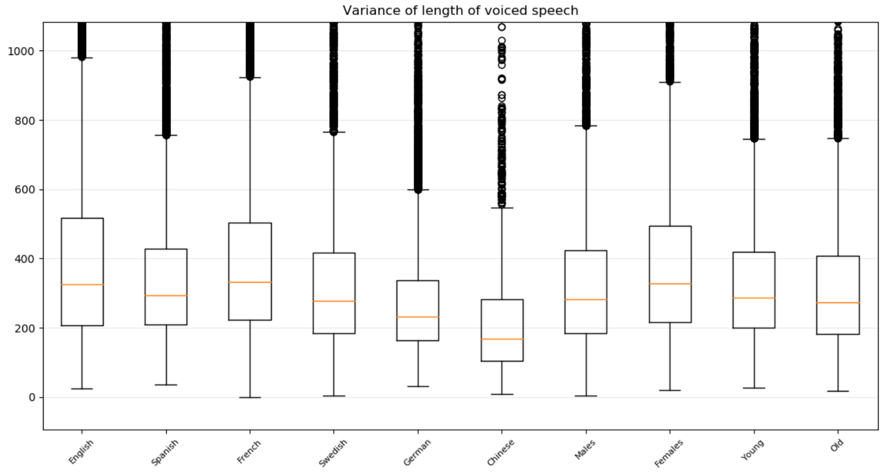


Figure A.4: Boxplots of the distribution of the variance of length of voiced speech (s^2) for each group.

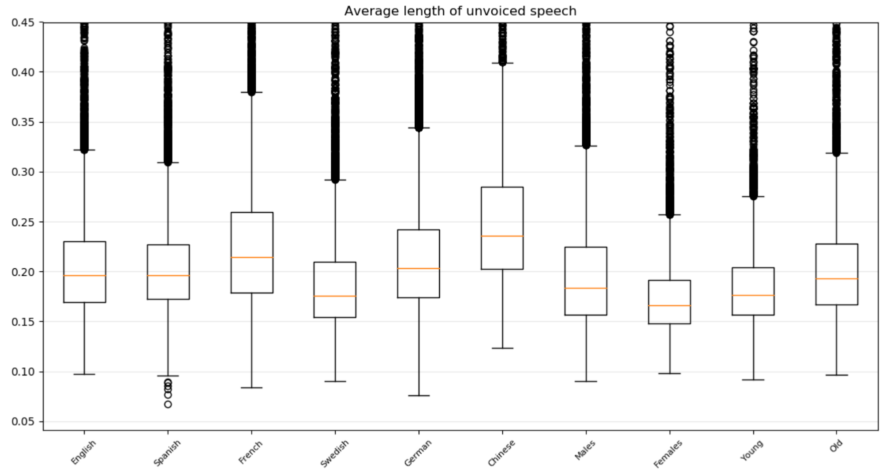


Figure A.5: Boxplots of the distribution of the average length of unvoiced speech (s) for each group.

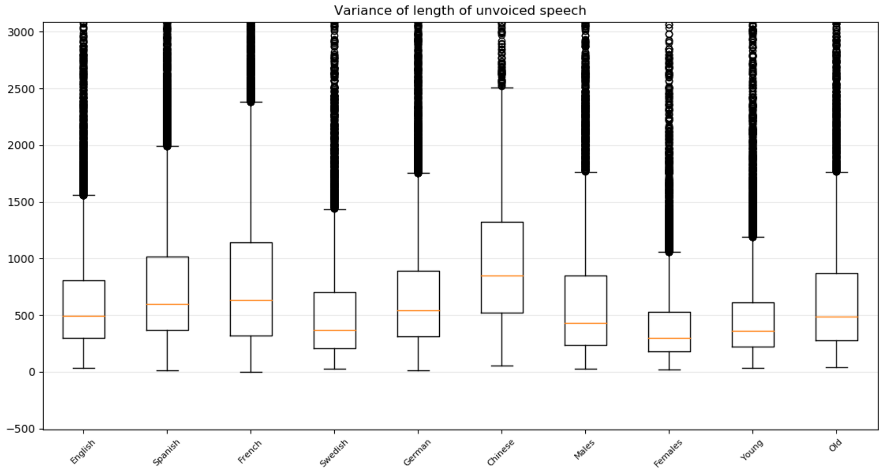


Figure A.6: Boxplots of the distribution of the variance of length of voiced speech (s^2) for each group.

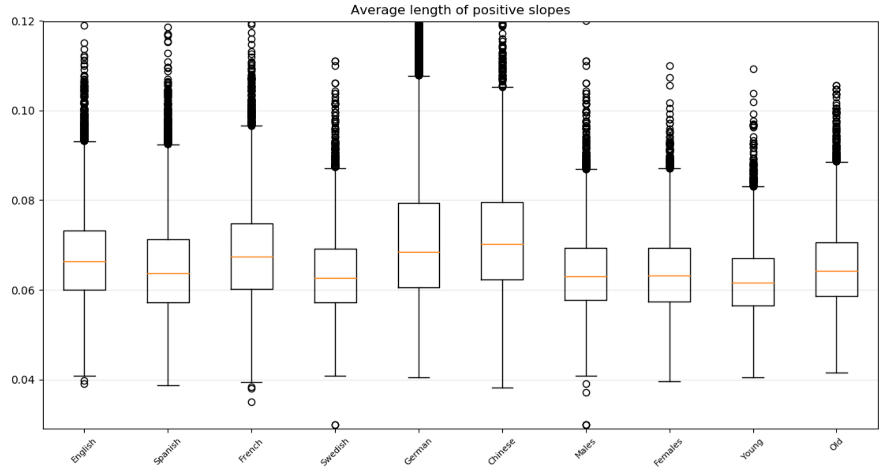


Figure A.7: Boxplots of the distribution of the average length of positive slopes (s) for each group.

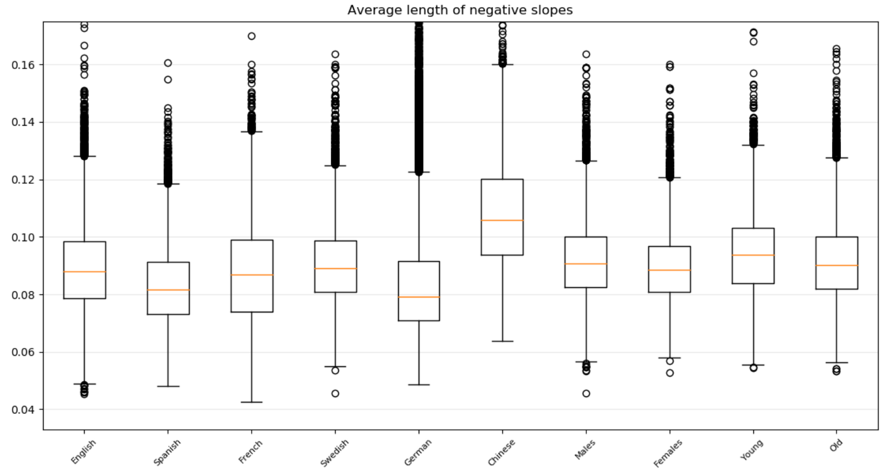


Figure A.8: Boxplots of the distribution of the average length of negative slopes (s) for each group.

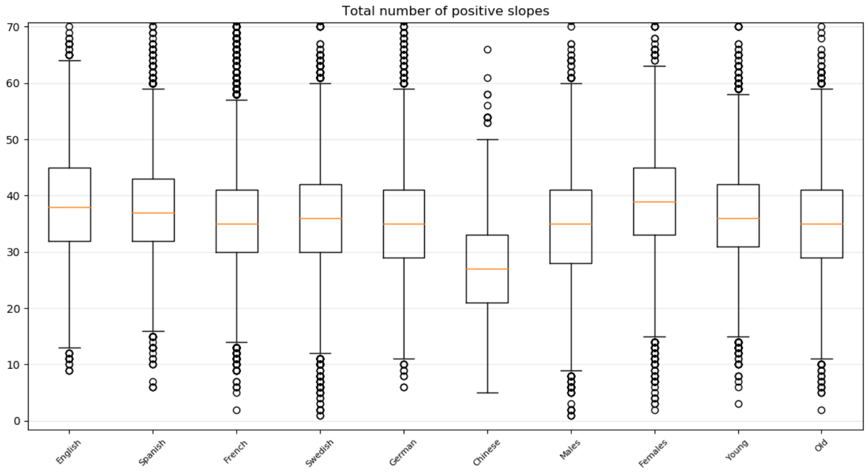


Figure A.9: Boxplots of the distribution of the total number of positive slopes for each group.

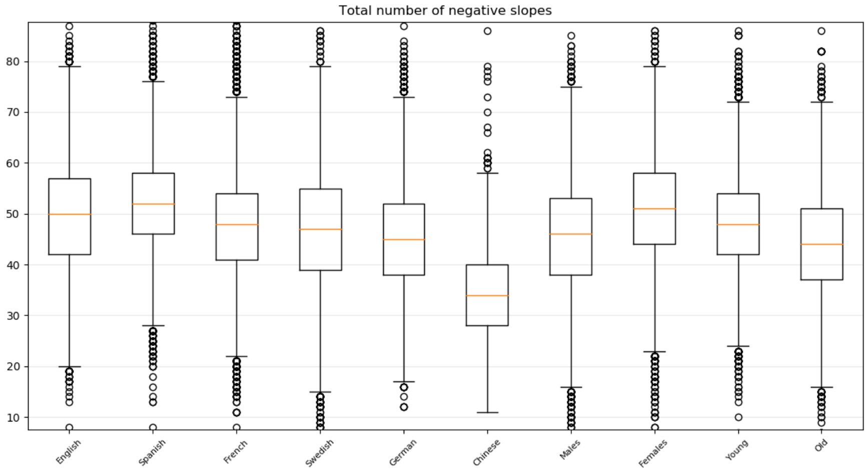


Figure A.10: Boxplots of the distribution of the total number of negative slopes for each group.

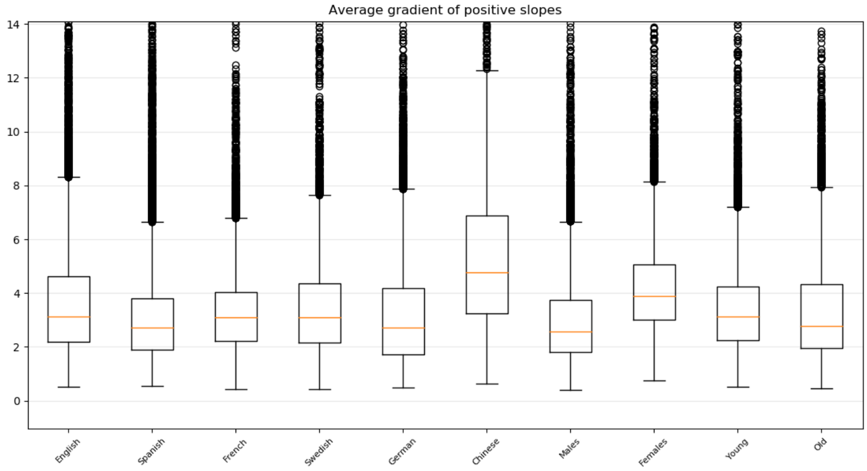


Figure A.11: Boxplots of the distribution of the average gradient of positive slopes (Hz/s) for each group.

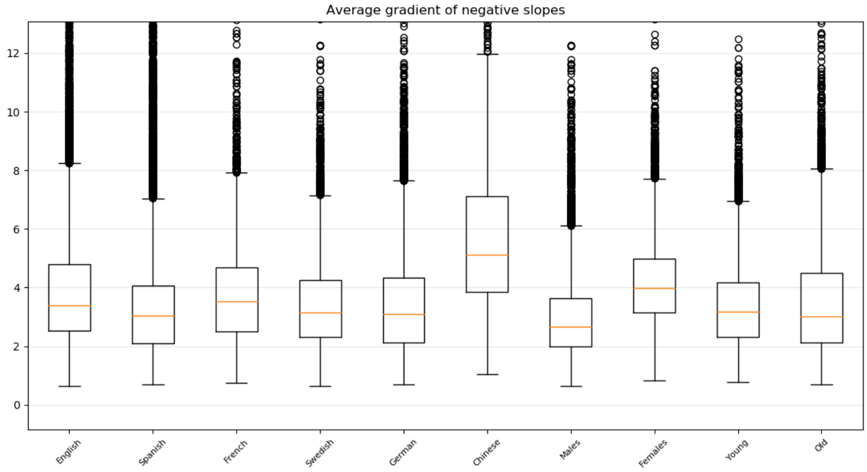


Figure A.12: Boxplots of the distribution of the average gradient of negative slopes (Hz/s) for each group.

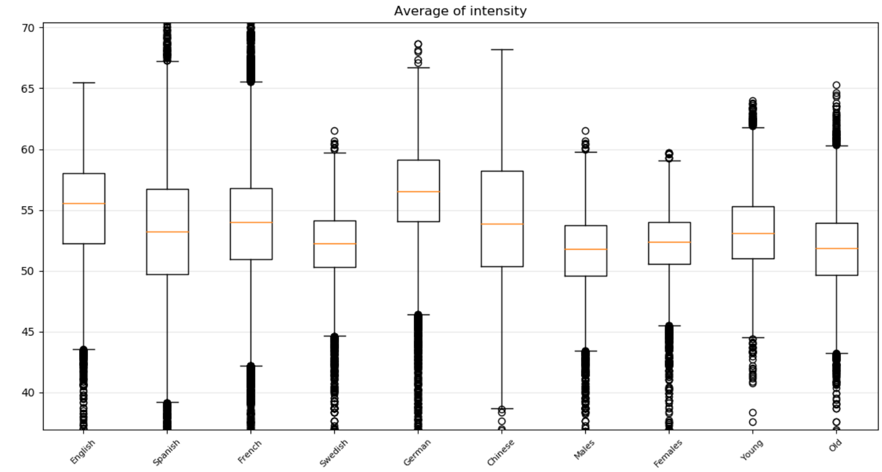


Figure A.13: Boxplots of the distribution of the average intensity (dB) for each group.

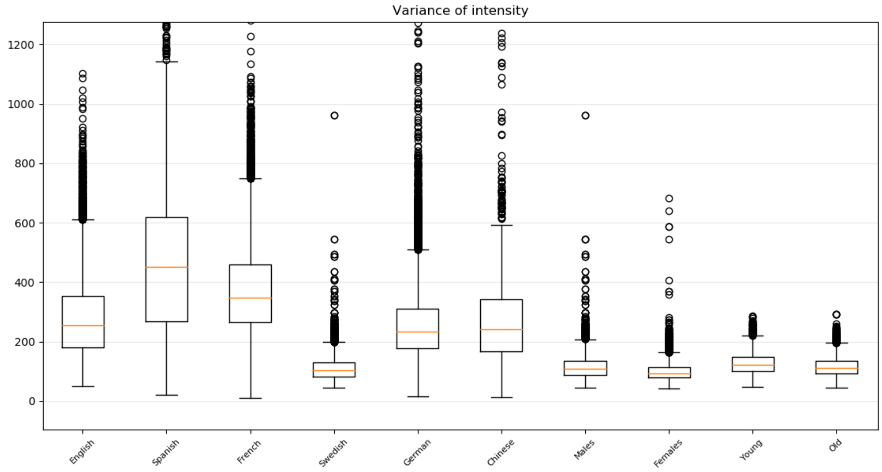


Figure A.14: Boxplots of the distribution of the intensity variance (dB^2) for each group.