B A C H E L O R ' S T H E S I S

Predicting Speaker Quality Using Embeddings

submitted by

Korbinian Koch

University of Hamburg

MIN Faculty

Department of Computer Science

Course of studies: Human-Computer Interaction (B. Sc.)

Matriculation Number: 6928742

Supervisor: Dr. Timo Baumann

Co-Supervisor: Prof. Dr. Chris Biemann

Licence

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this licence, visit
https://creativecommons.org/licenses/by-sa/4.0/.

Acknowledgements

I would like to thank everybody who supported me throughout my Bachelor's thesis, first and foremost my friends and family, who were always there when I needed them.

I would equally like to thank my supervisor, Timo Baumann, for helping to shape this thesis, providing me with the data and information I needed as well as the feedback and support throughout this work, even when asked for on a short notice.

My further thanks go out to my co-supervisor Chris Biemann, who did not just initiate this thesis with a personal invitation to collaborate, but also helped me navigate throughout the ups and downs of the project by finding the right words, and has supported me in many more ways throughout my studies.

Finally, I would like to thank Spotify and the coffee plant, without whom this thesis would not have been possible.

Abstract

A speaker's voice can significantly alter our perception of their authority, trustworthiness, or competence. A likable voice is able to attain and sustain attention, convince, sell, entertain and enhance our overall interaction experience. What exactly makes voices likable is however a product of countless factors, many of which are poorly understood even today. This thesis sets out to explore whether a nuanced likability score of a speaker's voice can be automatically determined independently from what the speaker is saying.

Speech samples were obtained from the Spoken Wikipedia Corpus and labeled using ranking information derived from crowd-sourced pairwise speaker preferences. As the amount of labeled data is very small, we will make use of transfer learning. For this, speaker embeddings pre-trained using generalized end-to-end loss (GE2E) and triplet loss (TRILL) are used and compared against traditional low-level audio features. All three feature types will be evaluated across Bi-LSTMs with attention, deep feed-forward neural networks, k-nearest neighbor regressors, and random forest regressors. To the best of our knowledge, we are the first ones to leverage embeddings for the prediction of likability scores.

We are analyzing the trained models in terms of their error, but also by how well they can deliver results that are specific to one speaker and independent of the spoken text. The best performing model overall is a k-nearest neighbor regressor trained on GE2E embeddings. Traditional audio features performed worse than embeddings in most conditions.

We conclude that speech embeddings do, within limitations, entail information about perceived speech likability and can be used for its prediction.

Contents

1 Introduction

1.1 Motivation

Voice user interfaces (VUI) are becoming increasingly popular. Both on phones but also on dedicated hardware like the Echo Dot or Nest Mini, voice assistants are able to help us manage simple tasks in our everyday lives. However, the main advantage of VUI, which is being a hands-free, eyes-free way of interaction, is also their biggest limitation. There are no visual cues that could for example signal the importance of a message. The expressiveness of the speech output is limited by its ability to convey non-lexical content through pronunciation, but also the inherent qualities of the voice itself. Since everything has to be conveyed with speech, speech output quality becomes an essential building block of the overall quality of a VUI.

However, due to the complex training of such voices, the most popular assistants currently offer no personalization options other than a few presets like accent (for example American vs. British English) one of two sexes, and of course language. While most assistants are able to learn to recognize their owner's voice quite accurately over time, all of them lack the individuality of having a unique voice themselves. Yet, it is likely that advances in speech generation and synthesis will make it feasible to change this in the future.

If these unique voices are just generated at random, the resulting voices will also include less likable ones, just like in reality. An automatic algorithm for speech likability prediction can be used to filter training data a priori or filter the generated voices a posteriori. It can also be used as an adversarial agent in the training process.

Humans are able to correctly attribute physical as well as behavioral qualities based on voice recordings. Not only are we able to assess the speaker's sex [1], age [2], race [3], height and weight [4], but also social status [5] and emotional states [6]. We are however not just able to assess these qualities, but listening to emotional speech alters our own emotions in a congruent way [7] and therefore shapes our perceived reality.

More complex features can as well be derived from speech. Humans are able to accurately match voice samples to their according facial photographs [8]. Computers have recently even been able to reconstruct facial images from just a short audio recording of that person speaking [9]. This serves as an example that human speech conveys a plethora of qualities where a single responsible property of the voice can likely not be pinpointed. We claim that speaker likability is among those qualities.

In previous works, the features used for speaker likability have exclusively been hand-selected acoustic low-level descriptors. The most used features include energy-related, spectral, and voicing-related descriptors as shown in Table 1.1, which are frequently extracted using OpenSMILE [10]. Features are often manually selected using a the-more-the-merrier approach and sometimes narrowed down using automatic feature selection methods [11].

The techniques used include support vector machines (SVMs) [13]–[16], decision trees [17] and random forests [12], genetic algorithms [15], restricted Boltzmann machines (RBMs) [18] and long short-term memory (LSTM) neural networks [19]. To the best of our knowledge, so far no one has explored predicting speaker likability directly from raw audio or using speech embeddings as intermediate high-level features.

This thesis builds on two previous publications [19], [20] by Timo Baumann, who is also the supervisor for this thesis. In the first [20], 5440 crowd-sourced pairwise likability ratings were collected for one fixed sentence by over 200 speakers from the German Spoken Wikipedia Corpus [21]. Given two speech samples, participants answered which voice they prefer. These ratings were later successfully predicted using a siamese Bi-LSTM network [19].

This approach has two limitations: first, it can only be applied to one spe-

4 energy-related LLD
sum of auditory spectrum (loudness)
sum of RASTA-style filtered auditory spectrum
RMS energy
zero-crossing rate
54 spectral LLD
RASTA-style auditory spectrum, bands 1–26 (0–8 kHz)
MFCC 1–14
spectral energy 250–650 Hz, 1 k–4 kHz
spectral roll off point 0.25, 0.50, 0.75, 0.90
spectral flux, entropy, variance, skewness, kurtosis,
slope, psycho-acoustic sharpness, harmonicity
6 voicing related LLD
F0 by SHS + Viterbi smoothing, probability of voicing
logarithmic HNR, jitter (local, delta), shimmer (local)

Table 1.1: Baseline feature set of the INTERSPEECH 2012 Speaker Trait Challenge with the subtask of likability prediction as shown in [12].

cific sentence seldomly heard in the wild, and second, it only allows for pairwise comparisons of two voices, but not the prediction of the likability of a single voice.

Since predicting likability per speaker instead of per rating reduces our training data from over 5000 to only ~200 unique samples, we employ the technique of transfer learning to our problem. Transfer learning refers to the technique of learning from one kind of problem, typically on a larger data set, and then applying the learned knowledge or representations to a different, but similar problem. In our use case, we will be using speech embeddings pre-trained for the task of speaker verification [22] and leverage them as input features for our prediction problem.

In speaker verification, the task is to verify the identity of a speaker from the characteristics of his or her voice. Due to the nature of this task, speech samples from one speaker will be close to each other in the embedding space, which makes

the learned embeddings especially suited for text-independent tasks like ours.

While speaker verification embeddings were not conceived as tools for transfer learning, TRILL embeddings [23] have most recently been presented as universal non-semantic representations of speech together with an according baseline for paralinguistic tasks. We will therefore put this claim to a test and evaluate if TRILL embeddings are indeed better suited for our paralinguistic task than other speech embeddings.

1.2 Problem Description

We want to turn the pairwise likability classification task into a single-sample likability regression task and predict a likability score from speech embeddings as input features.

Formally, we are looking for a function $h : X \mapsto [0, 1]$, that, given a speech input $x^i \in X$ returns the true degree of likability $l(x^i) \in [0, 1]$.

Given our aim to use a pre-trained n-dimensional embedding function $e : X \mapsto \mathbb{R}^n$, we are more specifically looking for a function $h' : \mathbb{R}^n \mapsto [0, 1]$, such that $h(x^i) = h'(e(x^i))$ is equal to the true degree of likability $l(x^i) \in [0, 1]$.

Additionally, we want our function h to be text-independent, meaning it should return the same output $h(x^i) = h(x^j)$ if the inputs come from the same speaker $speaker(x^i) = speaker(x^j), i \neq j$.

1.3 Research Question

The research question of this thesis is as follows:

> Can pre-trained speech embeddings replace manual features for speech likability prediction without significant change of prediction accuracy and—more importantly—make them text-independent?

Hypothesis Speech embeddings entail high-level speech descriptors and can be used as feature vectors for speech likability prediction. Since voice encoders trained for speaker verification produce similar and distinctive embeddings for the same speaker, predictions are independent of the spoken text.

1.4 Scope and Limitations

The scope of this project must be limited due to the fixed time frame of this thesis as well as some practical impediments. We have identified scope and limitations as follows:

- Speech embeddings can, in general, be applied to many tasks. However, the pre-trained embeddings have been trained mainly on English speech, while our data set contains only German speech. This will limit their descriptiveness and applicability.
- The collected likability ratings reflect the speaker preferences of a German and predominantly male subpopulation and will likely not generalize across cultures or genders. The same applies for the used speech data, which has predominantly been spoken by German men.
- The aim of this work is to provide qualified insight into the general aptness of speech embeddings for likability prediction and serve as a proof-of-concept for text independence.
- Due to the time-consuming nature of the model optimization procedure and the limited data, finding an overall optimally performing model is out of the scope of this thesis. We will however come up with recommendations based on our results, should one want to develop such a model.

2 Background

This section aims to equip the reader with the relevant theoretical background necessary to understand this thesis. In it, the following questions will be answered:

- What is audio and speech in the context of digital signal processing?
- How does speech likability fit into the area of paralinguistics and how are those two defined?
- What are the relevant learning techniques to solve our likability prediction task?
- What are and what characterizes the utilized speech embeddings?

In this section, related work will be referenced, discussed, and put into perspective where applicable.

2.1 Digital Signal Processing

A **signal** describes how some physical quantity changes over time and/or space [24]. Examples include sound waves, but also image data, temperature data and Electroencephalography (EEG) measurements. Analog, continuous signals, can be converted to **digital signals** by discretization, enabling them to be processed by a computer. The term **digital signal processing** describes all methods applied to digital signals with the goal of changing their characteristics or extracting information out of them [24]. In this thesis, we are looking at sound wave signals with the intent of extracting the information 'amount of likability' from it.

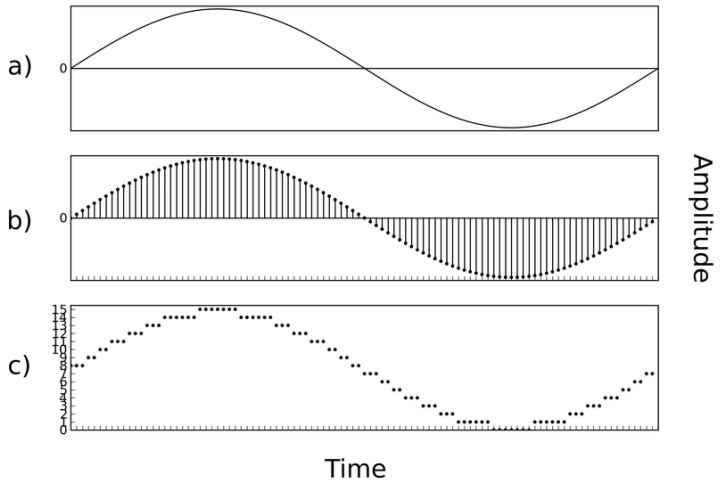

Figure 2.1: Discretization of an analog signal (a) just over time (b) as well as over time and amplitude (c), taken from [26].

2.1.1 Sound

Sounds are longitudinal pressure waves parallel to the application of energy and consist of zones of more and less tightly packed air molecules [25]. The alternations of high and low pressure cause vibrations in our eardrums, which are transmitted through the bones in our middle ear into the cochlea. Hair cells inside the cochlea respond to sounds of different frequencies, enabling us to distinguish between them. The movements of the hair cells are transformed into neural signals which are then sent via the auditory path into our brain, evoking the sensation of hearing [25].

2.1.2 Digital Audio

Sounds do not just cause vibrations in the eardrum, but all bodies of mass. Microphones make use of this and can convert the detected vibrations into an analog electrical signal. This signal can then be discretized into a **digital audio signal** using an analog-to-digital converter (ADC) [24]. An example of discretization can be seen in Figure 2.1.

The parameters defining the discretization process are the chosen resolutions

in time and space. The resolution over time, called sampling rate, is specified in Hertz (Hz) and imposes a natural limit to the highest possible frequency in the recording [24]. For high-quality recordings, a resolution of 44.100 Hz is often chosen, preserving all frequencies in the audible range between 0 and 22.050 Hz, but lower sampling rates are sensible as well, as only very few natural sounds actually are that high-pitched. The resolution over space chosen during quantization, also called amplitude or energy, is generally specified in bits, where for example 16 bit equals 65.536 possible amplitude values. The amplitude is determined by how far the microphone membrane is displaced during the recording and is directly related to the loudness of the signal. However, the conversion from measured displacement to digital signal (normalized between values of -1 and 1) is not reversible without exact knowledge of the recording parameters and the amplitude therefore unitless.

2.1.3 Speech

Speech can be defined as the transmission of language through the medium of sound produced by the vocal apparatus [27]. While language certainly starts in the brain, speech does not just originate from the mouth. It is rather produced by modulating air escaping through mouth and nose with assistance of numerous organs such as lungs, tongue, teeth, palate, lips or the vocal cords [25]. The plethora of involved organs, as seen in Figure 2.2, does not just result in a high expressiveness and adaptability of the voice itself ('intra-speaker variability'), but also considerable differences in the voices of different speakers ('inter-speaker variability'). Furthermore, since even the slightest change in the configuration of the vocal organs results in audibly different sounds, it is a phonetic truism that no one can say the same thing twice in *exactly* the same manner [28].

Due to the periodicity of the human voice, stemming from the repeated opening and closing of the vocal folds [25], speech can be quite accurately described as a sum of several sinusoids of different amplitudes and frequencies [24]. This enables spectral transformations such as the Fourier transform, which makes specific properties of the speech signal more accessible [27]. This method of pre-processing is therefore commonly employed when low-level and high-level features of speech are to be extracted. A visualization of the original waveform before and resulting

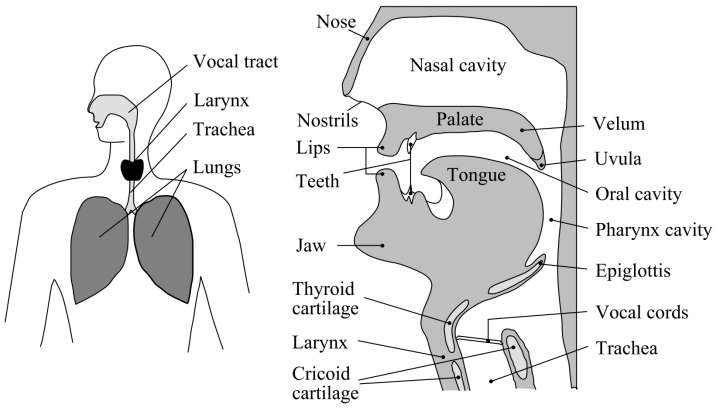

Figure 2.2: Organs for speech production, taken from [27].

frequency activations after the transformation, called spectrogram, can be seen in Figure 2.3.

2.2 Paralinguistics

Paralinguistics is the study of paralanguage and was first coined as a term by George L. Trager in the 1950s [29]. The *Cambridge English Dictionary* defines paralanguage as follows:

> **paralanguage:** the ways in which people show what they mean other than by the words they use, for example by their tone of voice, or by making sounds with the breath [30]

This definition implies that paralanguage has to be 'meant'. If, for example, a friend of yours interprets your tone of voice as you being sarcastic, but this was not intended by you, this would not count as an example of paralanguage under this definition. Vocal characteristics that do not carry any meaning at all, such as an innate lisp, would as well be excluded completely. The *Merriam-Webster.com* online dictionary, on the other hand, defines paralanguage as follows:

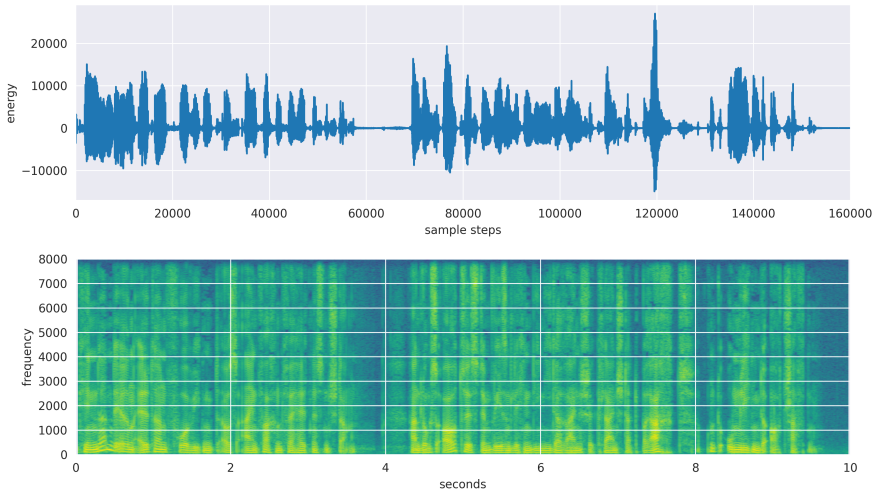

Figure 2.3: A 10 second speech sample from our data set, represented as waveform and spectrogram, where brighter colors represent higher energy density.

> **paralanguage:** optional vocal effects (such as tone of voice) that accompany or modify the phonemes of an utterance and that may communicate meaning [31]

This definition is much broader, as it does not require intention or meaning. Both the misread sarcasm as well as the lisp would be included under this definition. It is however highly misleading in the way that it defines paralanguage as an *optional* addition to the spoken language. It is impossible to imagine language spoken without paralinguistic features as 'all the properties of a voice as such are paralinguistic.' [32]

Definitions for paralanguage vary, just like the focus of paralinguistic research has shifted over time. In his original 1958 publication, Trager divides paralanguage into voice qualities, such as pitch, range and tempo, and vocalizations, which are specifically identifiable noises such as sighs, gasps, and mhm's [29]. Most recently, paralanguage has been defined as 'phenomena that are modulated onto or embedded into the verbal message, be this in acoustics (vocal, non-verbal phenomena) or in linguistics (connotations of single units or of bunches of units)' [33], which we shall accept as a definition for the purpose of this thesis. Examples of para-

linguistic traits and states given by the authors include gender, age, personality, intoxication, sleepiness, friendliness, mood, and emotion [33].

2.2.1 Computational Paralinguistics

Computational paralinguistics is the analysis and synthesis of paralanguage with computational means [33]. Research in computational paralinguistics takes many shapes and forms, but typically revolves around predicting paralinguistic qualities from labeled speech samples. The annual *Interspeech Computational Paralinguistics ChallengE (ComParE)* has so far hosted challenges ranging from the prediction of sleepiness from a given speech sample to the differentiation between different kinds of Styrian dialects [34].

2.2.2 Speech Likability

The question of whether and under which conditions speech is perceived as likable and how these perceptions can be predicted has been the subject of numerous paralinguistic studies, e.g. [17], [35]–[39]. While concepts such as voice attractiveness, voice likability, or even voice sexiness are not identical, there are also no sharp boundaries between them [40].

The 2012 *Interspeech Computational Paralinguistics ChallengE* [41] presented a likability sub-challenge, in which participants were asked to come up with techniques to predict the binary classes *likable* or *not likable* for short speech segments. The low-level descriptors proposed as useful input features were already presented in Table 1.1 in the introduction. The submissions for this challenge had an average recall between 54 and 65.8 percent [12], which can be interpreted as an indication that this kind of prediction is possible, but certainly not easy.

These results come as no surprise, as it is also hard for us to verbalize *what* makes a given voice especially likable, even though all of us have preconceptions of what a likable voice sounds like and are also able to manipulate our own voice to fit that conception. For example, while collecting utterances for a corpus of likable and unlikable Japanese speech, the authors of [42] identified that Japanese speaker

consistently elevated their fundamental frequency (f_0) when trying to sound more likable and lowered their f_0 when trying to sound less likable.

While ultimately being a mix of both, the likability of a given voice can be either mainly influenced by fixed factors or traits such as the speaker being born with a generally more likable voice or changing factors such as the speaker currently focusing on an especially clear pronunciation or (un)consciously manipulating their f_0. We shall name these two explanatory alternatives *likable trait hypothesis* and *likable state hypothesis*.

For the matter of simplification, we will treat the *likable trait hypothesis* as an axiom in this thesis. Since voice attractiveness has previously been linked to fixed traits such as greater bilateral body symmetry, including the larynx [43], as well as more attractive faces overall [44] we consider this assumption to be somewhat reasonable.

A comprehensive collection [40] of research on speech attractiveness, which also includes the work on the *Spoken Wikipedia Corpus* likability rankings [20] used in our experiments, has been published during the working time of this thesis and covers the evolutionary, social, and technological aspects of likable voices in much more detail than possible in this thesis.

In the context of our own research, **speech likability** will be defined as 'preference of one speaker over another given a particular speaking domain' [19], as this reflects the circumstances under which the likability ratings were collected. More on this can be read in Section 3.2.2.

2.3 Machine Learning

Machine learning is a sub-field of artificial intelligence and refers to a set of techniques that enable computers to complete tasks without being explicitly programmed and can improve their performance by 'learning' from data [45].

In predictive or **supervised learning**, we want to learn a function that maps inputs to outputs, in accordance with a set of input-output pairs called training data. Examples include the classification of hand-written digits from images or the

classification of spam emails. Descriptive or **unsupervised learning** on the other side aims to discover new knowledge about a set of data points—not input-output pairs. Examples include clustering or the discovery of latent factors or graph structure of the data [45]. The third type of machine learning is **reinforcement learning**, which we will not discuss in further detail.

2.3.1 Neural Networks

Artificial neural networks are universal function approximators consisting of interconnected artificial neurons inspired by the neurons in the human brain [46]. Given a collection of input features, neural networks will compute a collection of output features, which – in their most common form – depend on the internal structure, activation functions, weights and biases of the network. What makes neural networks useful is that they are able to learn, or, in other words, compare their computed outputs to desired outputs, calculate the resulting difference ('**error**'), and adjust their internal parameters in a way that makes the desired outputs more likely. In the mathematical sense, **learning** means to algorithmically approach the point in weight space under which our error function is minimal [46]. Typically, this is done by first computing partial derivatives for each neuron using the backpropagation algorithm, first developed in 1970 [47] and later named as such and popularized in 1986 [48]. Afterward, the weights of the neurons are adjusted using the method of gradient descent.

Neural networks are mainly used for **classification** tasks, where one input belongs to a subset of n classes, or **regression** tasks, where n continuous values have to be predicted from the input [45]. These two kinds of tasks are identical in the sense that their respective networks will have n output neurons, but differ in terms of their desired output. Whereas the desired output values for classification are either 1 or 0 and represent the membership of the respective class, the desired outputs for regression are real numbers, and could for example represent the height of a person or a probability between 0 and 1 [45].

Deep Neural Networks

Deep neural networks are a special type of neural network and commonly associated with the term 'deep learning' [49]. This term is misleading in the sense that the learning taking place in deep neural networks is following the exact same rules and steps as the learning taking place in shallow neural networks. Deep neural networks only differ in terms of their structure: They are 'deep' in the sense that they have **hidden layers** between the input and output layer and therefore a larger depth. This enables them to approximate functions of much higher complexity than shallow neural networks, as input values can be combined into meaningful higher-level features in later layers of the network [49].

Recurrent Neural Networks (RNN)

One of the biggest limitations of feed-forward neural networks is that they are only able to process inputs of a pre-defined length. While this works for many use cases, time series of an arbitrary length, such as audio signals, can not be processed with them in a way that preserves long-term dependencies. **Recurrent neural networks** solve this limitation by presenting the input recurrently. While classic feed-forward neural networks can only map from one input to the output, recurrent neural networks can thus draw from an entire history of inputs. In the context of audio signals, this means presenting only a short frame of a few milliseconds $x^{(t)}$ to the network and calculating an output $o^{(t)}$ and some hidden states $h^{(t)}$, which can be considered a lossy summary of the task-relevant history [49] or, in other words, a 'memory'. Then, the next frame $x^{(t+1)}$ and the hidden states $h^{(t)}$ are fed into the network again, until all frames of the signal have been presented. Typically, we are only interested in the last output, but not the intermediate ones.

Long Short-Term Memory (LSTM)

Recurrent neural networks have one serious drawback, which is that their gradients either explode or vanish over time (commonly known as vanishing gradient problem [50]), leading to weight oscillation or even no learning at all [51]. This problem is especially prevalent when working with long sequences and far-reaching

dependencies [50]. One way of remedying these shortcomings was the introduction of **long short-term memory** (LSTM) networks, conceived by Schmidhuber and Hofreiter in 1997, which introduced a new kind of recurrent network architecture combined with a compatible gradient-based learning algorithm [51]. The most significant changes that LSTMs introduced are the use of gate units, namely input and output gates, as well as an constant error carousel [51] and, in a later version, also forget gates [52], that enable the network to 'reset' itself.

Bi-LSTM

Bidirectional training refers to feeding sequential data into learning algorithm twice, once in a forward (left to right) and once in a backward pass (right to left). The two outputs can then be combined, for example through concatenation. This method has first been applied to LSTMs by Graves and Schmidhuber in 2005 for phoneme classification in continuous speech recognition [53], a use case related to extracting information from speech signals as well. Bidirectional LSTM networks are commonly abbreviated as **Bi-LSTM**.

Attention

Traditional LSTMs (or Bi-LSTMs, for that matter) are treating all input frames as equal. With respect to the predictive power of single frames, this is however unrealistic, as for example frames of an audio signal in which the speaker makes a pause will contribute much less to the perceived likability as frames in which they are audible. **Attention** mechanisms take advantage of this by 'attending to' some input frames more than to others. In essence, this works by learning the computation of attention vectors, which are multiplied with the inputs activation and determined by the frame itself as well as n contextual input frames, where n is also called attention width and defined a priori. All other attention parameters can be learned through backpropagation, just like the other parameters of the network.

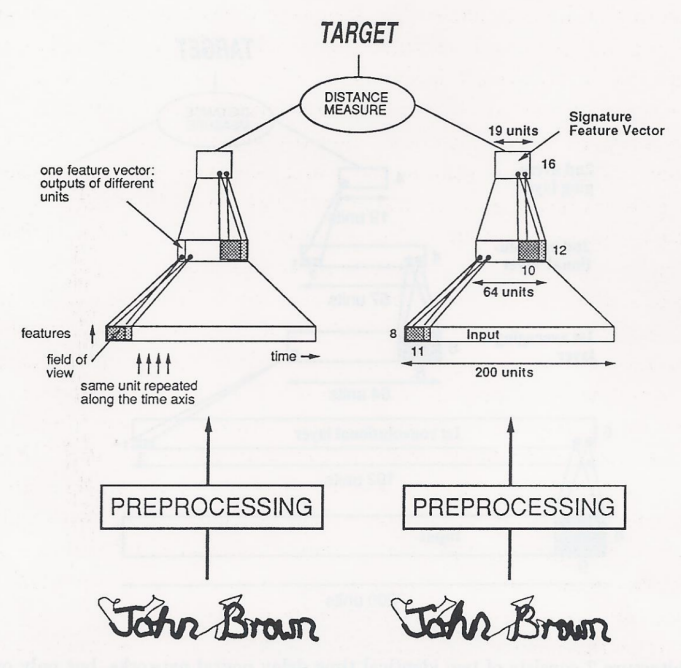

Figure 2.4: Siamese network architecture used by Bromley et al. [19].

Siamese Networks

Siamese networks consist of two identical networks joined at their output and were first introduced by Bromley et al. in 1993 [54]. They are used to measure relationships between pairs of inputs of the same type and share the same weights. For example, in their original publication Bromley et al. used them to compare two images of signatures with each other to decide if they come from the same person or not. The used architecture is visualized in Figure 2.4. Siamese networks are mainly defined by the choice of architecture used for the two sub-networks and the distance measure or, in other words, merging process.

As siamese networks are a perfect fit for pairs of inputs, a siamese Bi-LSTM has been used by Baumann [19] to reproduce the pairwise likability ratings collected in his previous study [20]. His architecture – see Figure 2.5 – differs from the one used by Bromley et al. to the extent that further learning takes place after computing the distance measure (here: subtraction). The utilized pairwise ratings will be discussed in more detail in Section 3.2.2.

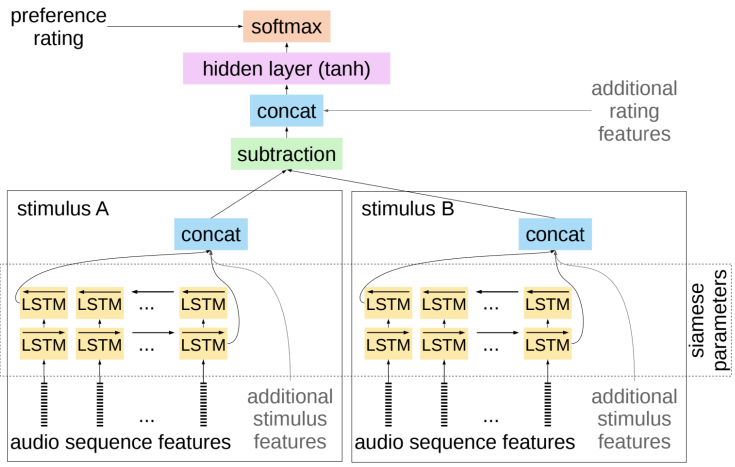

Figure 2.5: Siamese Bi-LSTM network architecture used by Baumann [19].

2.3.2 Transfer Learning

Training a model from scratch requires time, data, energy and money. While some of those variables can be increased easily (for example time: by waiting), others are sometimes impossible to increase even with unlimited resources (for example data on rarely occurring natural events). When large amounts of labeled data are not available, our models are likely to overfit, resulting in poor generalization.

Transfer learning exploits representations learned to solve a problem A in a different problem B that shares the same input type [49]. This is especially useful if we have access to significantly more training data for problem A than for problem B, as relevant representations of the same type of data are often similar and can be transferred from one context to another.

Transfer learning is commonly used in computer vision, where pre-trained models such as VGG16 and VGG19 [55], AlexNet [56] or ResNet [57] are frequently integrated as first layers of a model to rid oneself of the necessity to learn detecting basic features such as edges and corners from scratch. Likewise, natural language processing regularly employs pre-trained representations, for example using word2vec embeddings [58]. It is however much less common in speech processing, where the computation of traditional hand-crafted low-level descriptors (LLDs) is still most common and widely accepted pre-trained models have yet to

establish.

2.3.3 Random Forests

Neural networks are by far not the only machine learning methods. Classification or regression trees, also called **decision trees** learn a recursive partitioning of the data into increasingly smaller and more distinct subsets [59]. These partitioning decisions can be represented in the form of a tree, hence the name. Decision criteria and boundaries are determined in such a way that the entropy of the subsets decreases with each decision made. If, for example, your task is to determine whether a set of physical features belongs to a child or an adult, the weight will be much more informative than the eye color, and likely be selected as decision criterion (e.g. with a boundary of 56 kg). The same applies when the problem is a regression and not a classification task and used to predict the age. The final decision of the tree is determined by the value of the leaf a specific sample belongs to, e.g. a class or a numerical value. Decision trees can be fitted until all leafs only contain a single training sample or a given depth is reached [59].

As decision trees are prone to overfitting, their ability to generalize can be improved by pruning – the removal of less informative branches – or by training an entire set of decision trees, called a **random forest** [45]. To do this, subsets of the training data and subsets of their features are randomly selected to construct each tree. The resulting decisions of each tree can then be agglomerated into a single decision using the majority vote for classes or taking the average for regression [45].

2.3.4 k-Nearest Neighbors (kNN)

All previously discussed machine learning methods are so-called *parametric models*, meaning that that the training data can be 'forgotten' after all parameters – e.g. network weights or decision boundaries – of the models are learned. Another family of learning methods are *non-parametric models*, where the number of parameters is not fixed, but grows with the amount of training data [45]. Since these kinds of models rely on the availability of the training data (or a representation thereof)

even in the inference phase, they are also called *memory-based learning* techniques [45].

One such model is the **k-nearest neighbors** (kNN) algorithm. As the name implies, the kNN algorithm determines the k samples in the training set which have the closest distance to the input. Just like with random forests, the output is derived from the classes or values of these k samples. kNN is ultimately one of the simplest learning methods, as it only requires the definition of a distance measure and a value k and no further training.

While kNN is a very simple and versatile approach, it also has downsides: with an increasing number of input variables, the performance of kNN will quickly drop due to the *curse of dimensionality* [45]. It is also sensitive to outliers and uninformative features which do not help with coming to a right decision, but influence the distance measure just as much as highly informative features.

2.3.5 Model Validation

If we want to evaluate the performance of our model in an unbiased manner, we must measure its error on unseen data in order to approximate the error over all future data, called *generalization error* [45]. This is generally realized by removing a portion called testing or **test set** – usually between 10 and 20 percent – on which our model will be evaluated on. The remaining data used for training the model is called training or **train set**.

Using the test set for anything else than determining the final performance should be strictly avoided. However, the performance of most learning methods is determined by hyper-parameters, which have to be empirically determined. To be able to do this, the train set is further divided up into a new, smaller train set and a development or **dev set**. Models are then iteratively trained with different parameters on the new train set and evaluated on the dev set. For the final evaluation, the best performing parameters are selected and the model is trained again on both train and dev set data. Finally, the model is evaluated *once* on the test set. Typical proportions would be 80 percent train, 10 percent dev and 10 percent test data. Sometimes, the train, dev and test sets are also referred to as

train, test and validation set respectively.

The reliability of our error estimate can also be improved by not just calculating it on one, but k mutually exclusive test sets. This procedure is called **k-fold cross validation** [45]. In it, we are dividing our data up into k equal parts or 'folds'. We then construct k test-set-train-set pairs by taking one fold as test set and the remaining $k-1$ folds as training set until each fold has been used once as test set [45]. This implies that training from scratch has to take place k times, increasing experiment times manyfold.

The combination of hyper-parameter tuning and cross-validation is possible in two ways. One can either measure the performance of specific hyper-parameter combinations using cross-validation as inner loop (and not use dev sets), or cross-validate with hyper-parameter searches inside of each fold, where cross-validation forms the outer loop. Generally speaking, good hyper-parameters that were determined in one fold should be 'forgotten' and newly searched for in each following fold, as this knowledge was gained on data which is now in the test set. On the other hand, this way of parameter optimization is rarely seen in the wild, and popular frameworks such as *scikit-learn* also chose to only implement the first variant for combining cross-validation and parameter search, e.g. *GridSearchCV()*[1].

When data sets are very small, a random division into folds can result in unbalanced training and test sets, distorting the measured performance. **Stratification** of samples can make sure that each fold contains an equal proportion of samples of different value regions or classes. An early review of different cross validation and bootstrapping techniques suggested that 10-fold stratified cross validation may deliver the best basis for model selection, even when more folds are computationally feasible [60]. More information on stratification technique used in our experiment can be found in the methods section.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

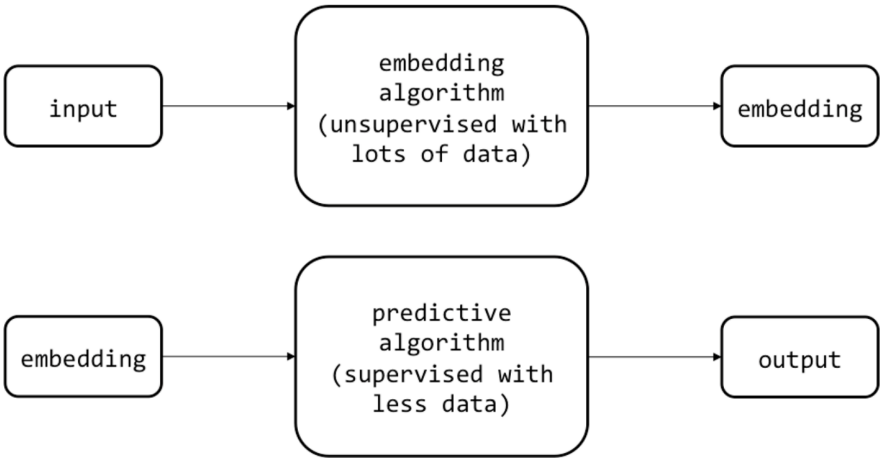

Figure 2.6: General process of learning and using embeddings, taken from [61].

2.4 Speech Embeddings

Learned representations are not only useful for transfer learning, but can also seen as a method of dimensionality reduction. There are many unsupervised techniques employed to reduce the dimensionality of data into compact representations, also called **embeddings**. Some of them, such as autoencoders, try to compress and then reconstruct inputs without additional information [49]. Others take the context of an input into account. The big advantage of both kinds of approaches is that they do not need any labels: the first because all desired information is in the input itself, the latter because contexts can be created at will. As long as data points can be divided into smaller parts, a contexts exist: sentences can be divided word by word, an image can be cut up into small tiles, and an audio recording can be cut up into shorter segments. Models can then learn to predict single data points from their context or vice versa. In the domain of speech, this could for example mean predicting an audio sample using two audio segments before and after that sample. If this is done using a deep neural network, the activations of one of the smaller layers can then be extracted and used as an embedding. The procedure of learning embeddings is represented by the upper part of Figure 2.6.

While the ability to predict from contexts or reconstruct compressed inputs may be useful in itself, the learned embeddings can also be exploited in related problems

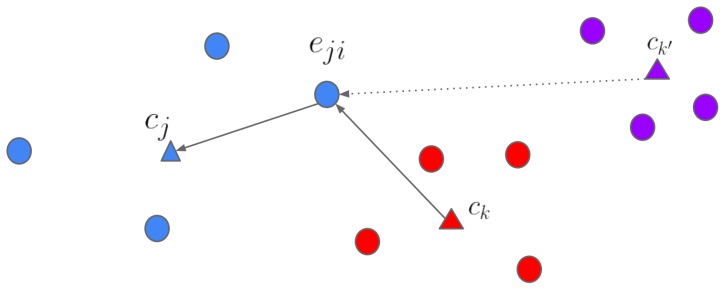

Figure 2.7: The pushing and pulling forces of contrast GE2E loss, visualized. Figure taken from [22].

using transfer learning. The exploitation phase is represented by the lower part of Figure 2.6. These secondary learning problems, which are usually supervised, now need significantly less training data, as meaningful and predictive features of the input data are already encoded in the embeddings [61].

While no universal **speech embedding** has established so far, many of them exist. In this thesis, we will make use of speech embeddings trained with two different kinds of loss functions, namely generalized end-to-end loss (GE2E) [22] and triplet loss (TRILL) [23], which will also be the names by which we reference the embeddings themselves in the remainder of this thesis.

2.4.1 GE2E

GE2E embeddings [22] are speech embeddings that were trained specifically for the task of speaker verification. In the speaker verification task you are trying to verify if an utterance belongs to a specific speaker based on known utterances of that speaker [22]. Applications include uses of the human voice as a biometric key, for example unlocking specific types of interactions with personal voice assistants, but also forensic use cases, where one has to determine if the voices of offender and defendant belong to the same person.

Generalized end-to-end loss builds upon tuple-based end-to-end loss (TE2E) [62], which has been made more efficient by focusing on difficult training instances [22]. During training, the model computes embeddings for $N * M$ utterances of

N different speakers, and tries to minimize the distance of each embedding $e_{ji}|j \in N, i \in M$ to the matching speaker centroid c_j while considering the distances to all centroids $c_k|k \in N$ represented by a similarity matrix. The implementation of GE2E loss using the softmax function ensures that embeddings are close to their correct speaker centroid while being maximally distant to all other speaker centroids, while the implementation using contrast loss only pushes the embedding away from the centroid of the most similar different speaker [22]. As the name implies, the training happens in an end-to-end fashion.

The embeddings are generated using a 3-layer LSTM network using stochastic gradient descent and have a projection size of 256 (for text-independent use-cases). Speech samples are not fed into the network directly, but first split up into frames of 25 ms each with a step size of 10 ms and 40-dimensional log-mel-filterbank energies are extracted as input features [22].

While the authors provided sufficient implementation details, they did not publish an open-source implementation of their model. Because of this, *Resemblyzer* embeddings [63] are used instead, which are a re-implementation of the described model. The model was trained on the LibriSpeech [64], VoxCeleb1 [65] and VoxCeleb2 [66] data sets, all of which include speech in the English language, while the latter two also contain other languages to some extent.

2.4.2 TRILL

While the embeddings for speaker verification were never meant to be used for transfer learning, the embeddings developed in the paper *Towards Learning a Universal Non-Semantic Representation of Speech* [23] have been created with specifically this purpose in mind. As the title suggests, the authors wanted to create a singular embedding type that performs well over all kinds of non-semantic tasks.

In comparison to GE2E loss, which compares samples with speaker centroids calculated from independent reference samples, TRILL uses temporal proximity as a self-supervision signal [23]. It constructs anchor-positive-negative triplets from a speech signal by sampling two contextual frames (anchor and positive)

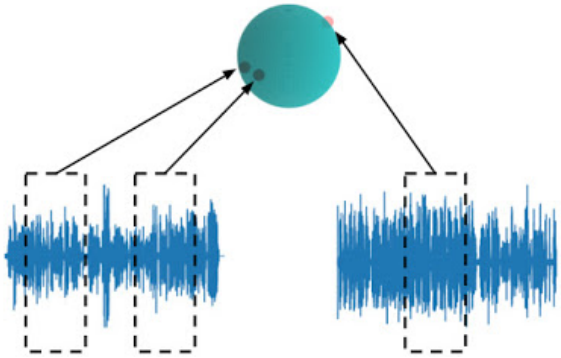

Figure 2.8: Anchor and positive will be closer to each other in embedding space than to the negative, taken from [67].

from the same recording and a negative frame from a different recording, as shown in Figure 2.8. Just like with GE2E loss, the distance towards the positive shall be reduced, while the distance to the negative shall be maximized. The advantage of this method is that it can also work with speakers for which only one recording exists or even without knowledge of the speaker at all, given that the probability of sampling two recordings of the same speaker purely by accident converges towards zero in a sufficiently large data set.

The TRILL embeddings are generated using a ResNet-50 convolutional neural network, using a series of convolutions, max-pooling and ResNet blocks, which finally result in a 2048-dimensional embedding. This embedding is not taken from the final layer, but the 19th layer of the model, which yielded the best results according to the authors [23]. The audio is split up into frames of 25 ms each with a step of 10 ms and converted to 64-channel mel-scale spectrograms as input features [23]. The model was trained on a subset of AudioSet [68], which contains speech extracted from 10-second YouTube segments and therefore all kinds of languages and speakers. The authors have publicly released their model, which is also used in this thesis. For inputs larger than ~180 ms, the model will output one embedding per time frame of that length, which enables the use of an LSTM for learning. TRILL embeddings were able to deliver state of the art performance on a number of paralinguistic classification tasks [23], which makes them seem especially promising considering that they were published only a few months ago.

3 Methods

This section provides a detailed overview of the experiments conducted in this thesis. It starts with a presentation of the data set and the applied cleaning and pre-processing steps. This is followed by a description of the developed model architectures. Ultimately the used evaluation criteria are presented.

3.1 Overview

The practical work of this thesis can be divided into two major parts:

First, the pairwise rating prediction by Baumann [19] was examined more closely. The performance of the presented approach was evaluated against a new baseline method in order to determine how it should be interpreted and what performance can be expected from traditional acoustic low-level descriptors. This set of inquiries will be summarized as **Experiment 1**.

Second, pairwise ratings were transformed into precise likability scores per speaker. The predictability of these scores was evaluated over two variables: three types of input features (acoustic features, GE2E embeddings, TRILL embeddings) as well as four different learning methods (Bi-LSTM, DNN, kNN, RF). This systematic investigation will be referred to as **Experiment 2**.

3.2 Data

This thesis builds upon two data sources: spoken Wikipedia articles and crowd-sourced ratings for their speakers. Both will be covered in the following sections.

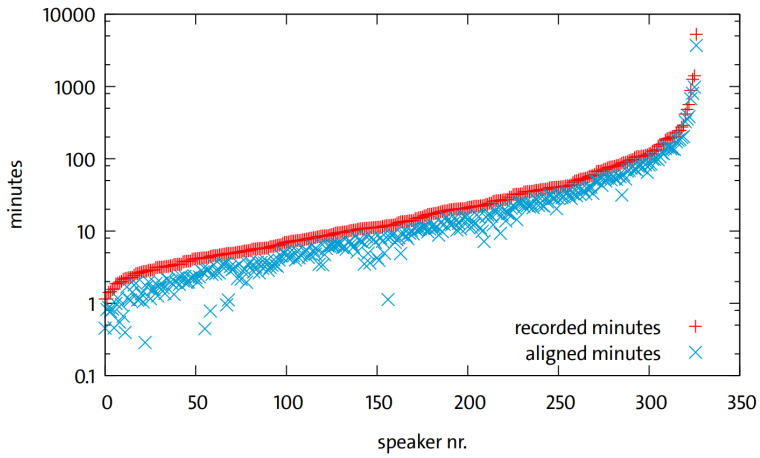

Figure 3.1: Speakers ordered by the amount of minutes they contributed to the German Spoken Wikipedia, taken from [21].

3.2.1 Spoken Wikipedia Corpus 2.0

The Spoken Wikipedia Corpus 2.0 (SWC) [21], or – more precisely – the Spoken Wikipedia Corpora, are a collection of spoken Wikipedia articles in the German, English and Dutch language. The German corpus, which is used in this thesis, consists of 1014 articles by 339 speakers with a total duration of 386 hours [21]. Articles cover a wide variety of topics, ranging from famous personalities over geographical areas to technological concepts.

While the average number of articles spoken per speaker is around three, the actual amount of contribution varies heavily among speakers, as can be seen in Figure 3.1. The most active speaker contributed a staggering 60 hours of speech, while others only contributed a few minutes [21].

Even though all articles are time-aligned word by word to their respective text, we will only be using the audio recordings of the corpus on their own. The reason for this is not that we are striving to build a text-independent model: Alignments themselves are just another feature type that can be generated without previous knowledge of the spoken text utilizing any speech-to-text framework as intermediary step. It is rather that they are simply not necessary to create embeddings or audio features, which are generated from the audio file alone.

It is both conceivable and likely that an approach fusing audio and text together will yield better results. Inclusion of phonemes into the model was able to boost performance considerably in the experiments done by Baumann [19]. In this work, however, we are deliberately limiting ourselves to use audio and nothing else, as we are not interested in building the best possible model, but to find out if and how much likability information is entailed in speech embeddings alone.

All audio files in the corpus are stored in the Ogg Vorbis (.ogg) format with varying channels (mono vs. stereo), a sampling rate of 44,100 Hz and varying bit rates. All bit rates are sufficient to convey a speech signal in high quality and will be equalized later on.

3.2.2 Ratings

In a follow-up study, Baumann called for voluntary participants on the web to give binary likability preferences for pairs of speakers from the German SWC [20]. This practice of engaging a diverse group of volunteers or 'crowd' to gather data online instead of just a narrow group of participants in a controlled lab setting is referred to as crowd-sourcing [69].

While one could argue that crowd-sourced likability ratings are less reliable than ratings collected in a laboratory setup, it has been shown that despite measurable differences, both are highly correlated (Pearson's $r = 0.92$, $p < 0.001$) even with inclusion of unreliable participants and are therefore valid measurements [69]. As likability ratings always involve a subjective component, a perfect correlation can not be expected.

The precise question Baumann asked his participants is *'Von welcher Stimme würdest Du lieber Wikipediaartikel vorgelesen bekommen?'* ('Which voice would you prefer to have Wikipedia articles read by?') [70]. Even though the word 'likability' has never been mentioned towards the participants, the survey measures speech likability as defined by us in Section 2.2.2. In simpler words: It measures whatever it is measuring and we call the unit 'likability'. While this definition is certainly circular, it does capture likability in the form of positively assessed and preferred **speaker quality**, which is also the term chosen for the title of this

thesis.

In order to reduce the influence of any confounding variables, Baumann asked the participants to ignore the recording quality of the used audio device and focus on the voice alone. Furthermore, only speech segments with identical text were shown to the participants. In particular, the sentence *"Sie hören den Artikel [...] aus Wikipedia, der freien Enzyklopädie"* ("You are listening to the article [...] from Wikipedia, the free encyclopedia") was used, and the article name was replaced by noise [20]. We will refer to speech samples in this standardized form as **anonymized samples**.

Since speech segments had to consist of this exact sentence, only 227 samples were included in the rating procedure [20], coming from 215 unique speakers. Reasons for exclusion included the speaker saying something else or failed automatic alignment and removal of the article name.

Besides the ratings, metadata on the raters in form of age, sex and dialectal origin were collected [20]. However, ratings do not just depend on personal preference, but are to some degree random, which is especially true if two speakers sound very similar. The rating procedure can therefore be represented in the form

$$S = \{s_1, \ldots, s_{227}\}$$
$$P = \{p_1, \ldots, p_n\}$$
$$r_{(p_k,\epsilon)} : S \times S \mapsto \{0, 1\} \mid p_k \in P$$

where S is the set of anonymized samples, P is the set of participants and $r_{(p_k,\epsilon)}$ is a functional preference model that depends on the participant p_k and some random error ϵ and returns either a 0 or 1, depending on which sample is preferred. A single **rating** then takes the form

$$r_{(p_k,\epsilon)}(s_i, s_j) \mid p_k \in P; s_i, s_j \in S; i \neq j$$

The sample pairs presented to participants were not randomly selected, but some preference was given to pairs that resulted in a high number of conflicting preferences, meaning there was high disagreement among raters which speaker is

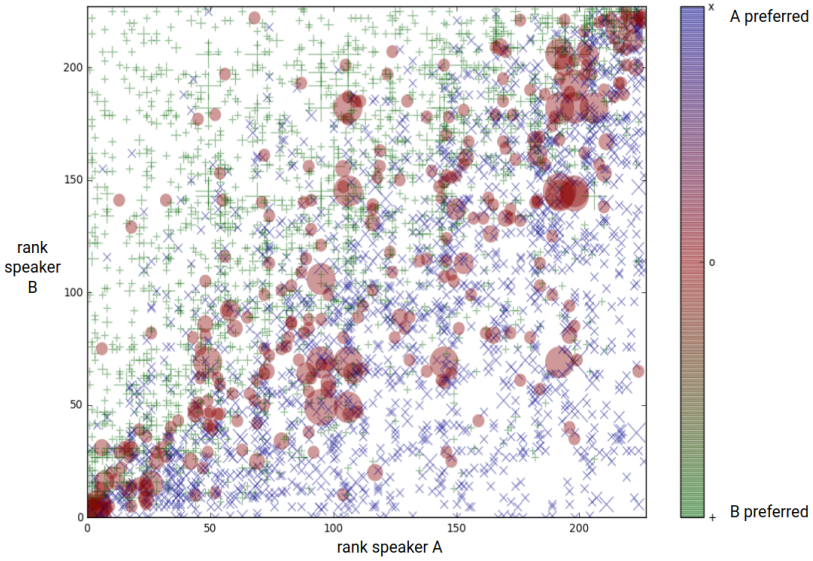

Figure 3.2: Scatter plot of rated pairs, ordered by rank and color coded by the average preference. Larger markers represent a higher number of ratings. Adapted from [20].

more likable. This should ensure that the information gained from each rating is as high as possible, as we will not gain much by gathering more evidence for hierarchies on which we are already very certain. However, this also means that the influence of ϵ will be larger for those ratings.

In total, Baumann collected 5440 ratings from 168 participants [20].

3.2.3 Ranking

In order to turn the collected ratings into a global ranking, the Microsoft TrueSkill algorithm [71] was continuously applied to the gathered ratings. Since TrueSkill was developed for up-to-date gaming scoreboards and thus weights more recent scores stronger, the order of all ratings was permutated and multiple rankings calculated. Then, a final **ranking** from worst to best was determined by calculating the median rank (e.g. 146 out of 227) for each anonymized sample [20].

Of course, not all used ratings are agreeing with each other in terms of their

preference. A visualization of ratings in relation to ranks can be seen in Figure 3.2. Frequent disagreement can be observed especially between samples with similar ranks. However, the ranks are plausible on a larger scale, despite the intra-rater and inter-rater inconsistencies.

The pairwise ratings from [20] were acquired on a single sentence and therefore only convey qualified information about that particular segment of speech. It is however likely that speakers do not considerably change their style of speaking during one recording. Changes between recordings could be observed, as some speakers for which two anonymized samples were included in the study ended up with ranks further apart than expected. For example, two anonymized samples of the speaker *zumorc* ended up with ranks of 51 and 151 respectively. The ranks of all duplicate speakers can be seen in Table 3.1. Please note that specific ranks can appear twice, as they are medians.

speaker	ranks
anhezu	106, 186, 190
barfisch	173, 205
buecherfresser	143, 179
diekeimzelle	36, 45
footox	80, 169
martinschlederer	109, 193
msh	120, 143
pavl90	70, 179
phantom	187, 193
souffleuse	169, 197
zumorc	51, 151

Table 3.1: Ranks of the speakers of which anonymized samples were included more than once.

The average individual rating distance for those speakers is 54.5, which represents roughly a quarter of the available range from 0 to 226. While this is evidence against our *likable trait hypothesis*, it shall be noted that these duplicates only exist because data from the small-scale pre-study conducted by Baumann [20]

Figure 3.3: Typical contents of an SWC article folder.

accidentally made its way into the actual study. Yet, for us, it is not reliably reconstructable how valid these ratings were in the first place or how seriously these differences should be taken. For example, when listening to the two samples by the user *footox*, there is almost no audible difference at all, despite both ranks being 89 apart.

Unfortunately, the existence and scope of these shortcomings has not been detected in the original publication and only been fully understood by us after our own experiments have been conducted. No measures have been taken to compensate for the spread in ranks. While for each speaker only *one* consistent rank has been used in the final experiments, *which* of the multiple ranks was selected was determined purely by chance. Luckily, no contradictory training samples were produced and only a small number of samples were affected. Nonetheless, this mistake can without a doubt be used to bring the reliability of our results into question.

3.2.4 Cleaning and Pre-Processing

The German corpus contains 1014 article folders, but not all of them are consistent in terms of their content and usable for our experiments. A typical article folder will consist of seven files, as seen in Figure 3.3, where audio.ogg is the audio recording of interest. Information on the speaker is stored in info.json.

In order to be able to use the data effectively for our purposes, we are doing the following pre-processing steps:

For each article folder:

1. Check if the contents are complete
2. Determine the speaker name

3. Concatenate all audio files, if there are more than one
4. Cut out 10 samples of 10 seconds each from the audio file and store them as .wav files with one channel, 16.000 Hz and 256 kbps.
5. Look up the speaker's rank (if known), divide it by 227 in order to squeeze it between 0 and 1, and include it in an external dictionary

During the pre-processing procedure, several articles could be identified as not usable, as can be seen in Table 3.2.

article folders	1014
of which: not empty	1012
of which: with speaker name	999

Table 3.2: Reduction of usable articles during pre-processing.

Just as for the articles, the number of usable speakers is also lower than suggested by the corpus parameters, as seen in Table 3.3. There are two reasons for this: naming errors and exclusion from the crowd-sourced rating procedure.

For example, it is quite likely that the speakers *butchm* and *derurheberdieser-audiodateibinichbutchm* ('theoriginatorofthisaudiofileamibutchm') are identical, but since spotting and correcting these kinds of mistakes on a large scale would have required a lot of manual labor, speaker names have been left unchanged. While a certain number of the unrated speakers may have been unidentified duplicates, their expected count is likely low. The larger portion of speakers without known rank can be explained by exclusion from the original crowd-sourced rating experiment, which included only 215 unique speakers.

unique speakers	333
of which: with known rank	215
of which: used in Experiment 2	199

Table 3.3: Reduction of usable speakers during pre-processing.

The difference between 215 and 199 can be explained by a fault in the encoding

of diacritics in the speakers' names, which were used for checking if they had a known rank. Due to this, some of the speakers were treated as if their rank was not known and excluded from the experiment, even though one existed. Sadly, this has been detected too late in the project to be fixed, and the number of included speakers is therefore around 7 percent smaller than it had to be.

3.2.5 Feature generation

For each .wav file, three types of features were generated: one GE2E embedding, 54 TRILL embeddings (one every ~185 ms) and 1000 data points each for 50 different audio features.

The acoustic features are identical to the ones extracted by Baumann and include mel-frequency cepstral coefficients (MFCCs), the fundamental frequency f_0, fundamental frequency variation (FFV), jitter, shimmer, and harmonics-to-noise (HNR) ratio [19]. Additionally, linear filter bank (LFBANK) coefficients were generated. The features were generated using a sliding window with a stride of 10 ms. It should be noted that the acoustic features will not be used without further processing, such as mean-aggregation.

directory	size	dimensionality per sample
original German corpus	15.2 GB	
selected .wav samples	3.1 GB	$10\text{s} \times 16,000\text{Hz} = 160,000$
GE2E embeddings	17.2 MB	$1 \times 256 = 256$
TRILL embeddings	4.3 GB	$54 \times 2048 = 110,592$
audio features	7.5 GB	$1000 \times 50 = 50,000$

Table 3.4: Sizes of data sets and features.

Contrary to what one might expect, all features except for the GE2E embeddings are larger than the audio files they originate from, as can be seen in Table 3.4. Nonetheless, all features are reducing the dimensionality of the data over time and space, even if just slightly.

Figure 3.4: The pre-processing pipeline from original corpus to used features.

3.3 Experiment 1: Pairwise Classification

Baumann successfully predicted the collected ratings from [20] using a siamese Bi-LSTM model in a later publication [19]. The model architecture has already been shown in Figure 2.5 in the backgrounds section on siamese models. The used input features were comprised of the audio features previously described, this time only calculated solely on the anonymized samples and not using the log-frequency bank features. In order to avoid the problem of vanishing gradients, the audio features were aggregated over the time dimension, taking the average of 5 consecutive frames each [19]. In one condition, phone embeddings for the phonetic alignments from the Spoken Wikipedida Corpus were used as well.

The purpose of the experiment was to find out if the pairwise preferences could be predicted from the features. Since the training data includes contradicting ratings, a prediction accuracy of 100 percent is practically unachievable. However, Baumann determined that for example ratings from male and female participants

only showed a moderate correlation ($\tau = 0.44$, $p \ll .001$, Kendall's Tau), explaining different preferences between those groups [20]. It is therefore reasonable to ask the question whether a good enough prediction accuracy could be achieved under consideration of these additional meta-features.

3.3.1 Test Sets

For evaluation, Baumann constructed three different kinds of test sets, namely:

naïve: take out 100 random samples from the data set

easy-0.25: take out 100 random samples from the data set of which the samples are at least 57 ranks (25 percent) apart

easy-0.5: take out 100 random samples from the data set of which the samples are at least 114 ranks (50 percent) apart

These test sets should reflect different degrees of difficulty for the model, ordered from hard to easy. At first, this choice seems sensible, but it has a significant downside: All three test sets are made up of anonymized samples that may also included in the training set. This means the model will likely have seen both of the inputs before and just memorized them either together or individually. Even though it is unlikely to encounter completely identical inputs (including age, sex and dialectic origin), this dramatically reduces the validity of the measured accuracies, as it is practically impossible to detect overfitting.

For comparability, we will re-use the test sets from before, but additionally construct a fourth one:

extern: take out *all appearances* of ratings involving a randomly selected speaker until you exceed 100 ratings

This test set will allow us to determine the performance on inputs of which at least one of the inputs has never been seen before. This condition is still not ideal, but far more reliable than the other three test sets.

3.3.2 Models

We will re-run the experiment by Baumann using code provided by the author. Both models are trained for 50 iterations using the best performing parameters reported in the paper.

Additionally, we will construct a new baseline method, which we will call **rank-based classifier**. It does not 'listen' to the audio at all and only knows ranks as input. It works as follows:

1. Calculate the ranks again using the remaining ratings in the training set
2. When classifying the preference, always let the higher rank win
3. If the rank is unknown (e.g. in extern), pretend the speaker is mediocre (rank 114/227)

A very similar classifier has been constructed by Baumann in his publication on ranks [20], where it achieved an accuracy of 68 percent using 100-fold cross-validation. It was however not evaluated on the above mentioned test sets from [19].

The experiment will be run on our local machine, the parameters of which can be seen in Table 3.5 further below. For reference, we will also report the results from the original publication [19].

3.4 Experiment 2: Regression

The prediction of pairwise comparisons enables us to learn something about the influence of individual raters, but does not enable us to make assertions about the degree of likability of individual speakers. Of course, one could evaluate a new speaker against many speakers with known ranks and determine where in the hierarchy it 'fits in', but this was not what the model was trained for and will introduce even more uncertainty.

We will turn the problem into a regression task, meaning that we are taking only one speaker as input and are predicting a degree of likability. This degree of

likability will take the form of a value between 0 and 1 and is derived from the rank of the speaker, where the worst speaker has a likability score of 0 percent and the best speaker a score of 100 percent. This of course assumes that there is no speaker out there that is more or less likable than the best or worst speaker in our data set, which is a simplification we are willing to make.

The disregarding or ratings in favor of ranks makes it impossible to consider rater metadata for prediction. This is exactly what we want: not to learn about preferences of individual raters, but to learn the 'true' likability of a speaker.

In order to achieve this, we are systematically comparing the performance of four different learning methods across three feature types. The features will be the acoustic features from Experiment 1 as well as GE2E and TRILL embeddings. The learning methods we will compare are a Bi-LSTM with self-attention, a feed-forward neural net, k-nearest neighbors and random forests. As some of the learning methods are not able to process time series, the respective features (acoustic and TRILL) will be compressed in the time dimension by averaging for those methods.

3.4.1 Models

The architectures of the used models will have to change with respect to the input features, as they have different dimensions. In the next segment, we will describe the exact make-up of our models and the hyper-parameters we will consider.

Feed-Forward Deep Neural Network

One feed-forward neural network was constructed for each input type. The input layers were built to fit the size of the compressed features, having a size of 50, 256 and 2048 neurons respectively. They are followed by 5 fully connected hidden layers for the smaller audio features and GE2E embeddings, and 10 fully connected hidden layers for the large TRILL embeddings. The fully connected output layer has only one neuron and should output the likability score. A visualization of the networks with hidden layer sizes can be seen in Figure 3.5.

All networks were trained using mean squared error as loss function, adamax as optimizer, tanh as activation function and a batch size of 20. Learning rate and dropout rate will be determined by random grid search, meaning that only some combinations are tried out (learning rate: 5e-7, 5e-8, 5e-9; dropout: 0.5, 0.25, 0). The number of iterations will be determined by early stopping, but is limited to a maximum of 100. All networks were implemented as sequential Keras[1] models, a framework built on top of TensorFlow[2]. Model tracking will be performed with weights and biases[3].

Bi-LSTM with Attention

The Bi-LSTM models are built on top of the developed feed-forward neural networks. As only audio features and TRILL embeddings are available as time-series, GE2E embeddings will not be used in this condition. In order to make the length of the time-series more similar, the audio features were aggregated, always taking the average of 20 consecutive frames. This reduces the number of time steps from 1000 to 50.

A bidirectional LSTM layer (merged by concatenation) was added in front of the first feed-forward layer, followed by a self-attention layer with an attention width of 15 frames (corresponding to roughly 3 seconds of audio). As implementation of an attention mechanism, the *keras-self-attention* package[4] was selected, mainly because of its ease of use. A visualization of the architecture of the Bi-LSTM part of the models can bee seen in Figure 3.6.

The exact same hyper-parameters as for the feed-forward neural networks were used. As grid search was not feasible for the Bi-LSTMs (with a single run taking up to 34 hours), the best performing parameters from the feed-forward neural networks will be used for learning rate and dropout.

[1] `https://keras.io/`
[2] `https://www.tensorflow.org/`
[3] `https://wandb.ai/`
[4] `https://github.com/CyberZHG/keras-self-attention`

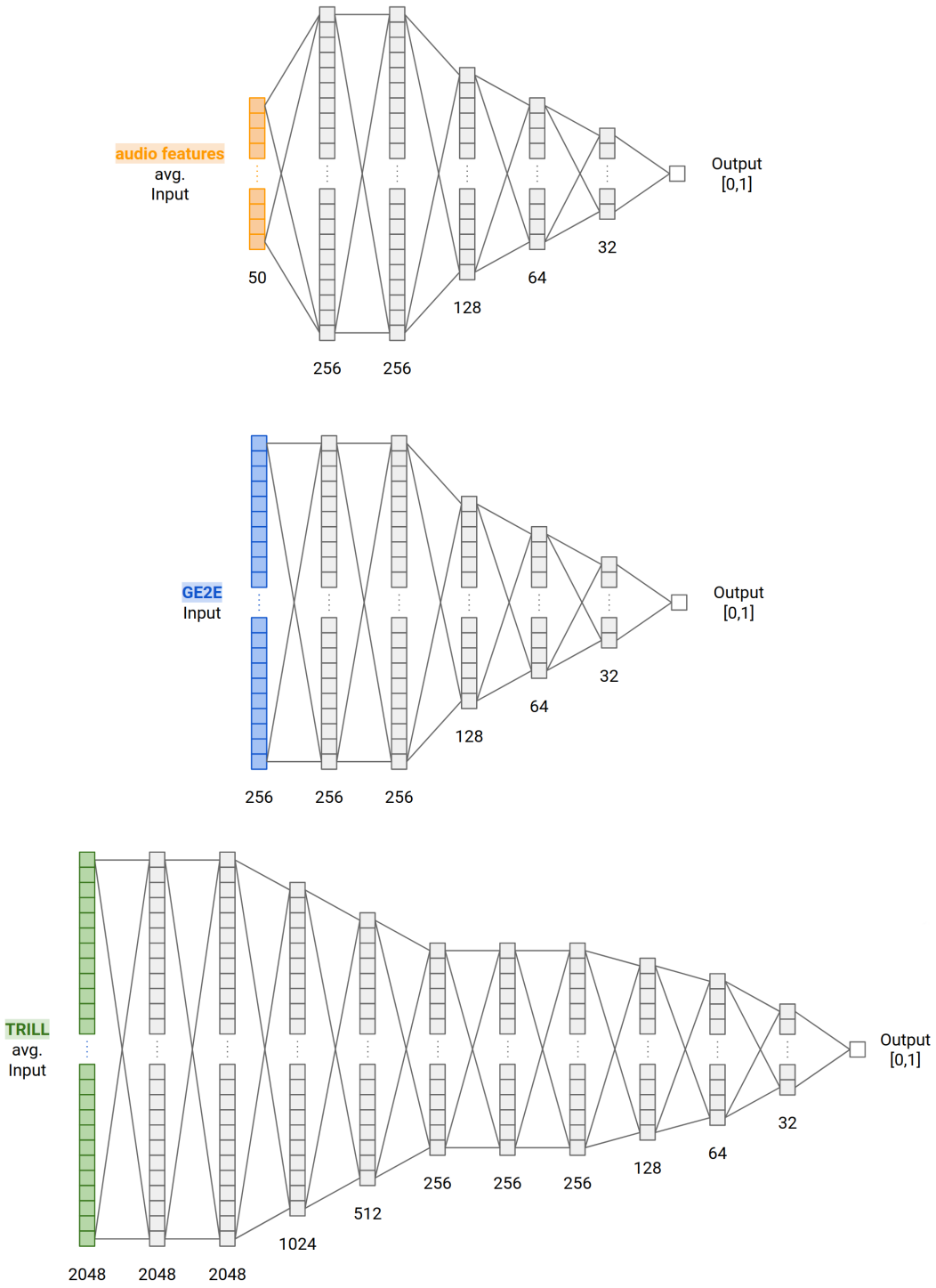

Figure 3.5: The architectures of the three feed-forward neural networks.

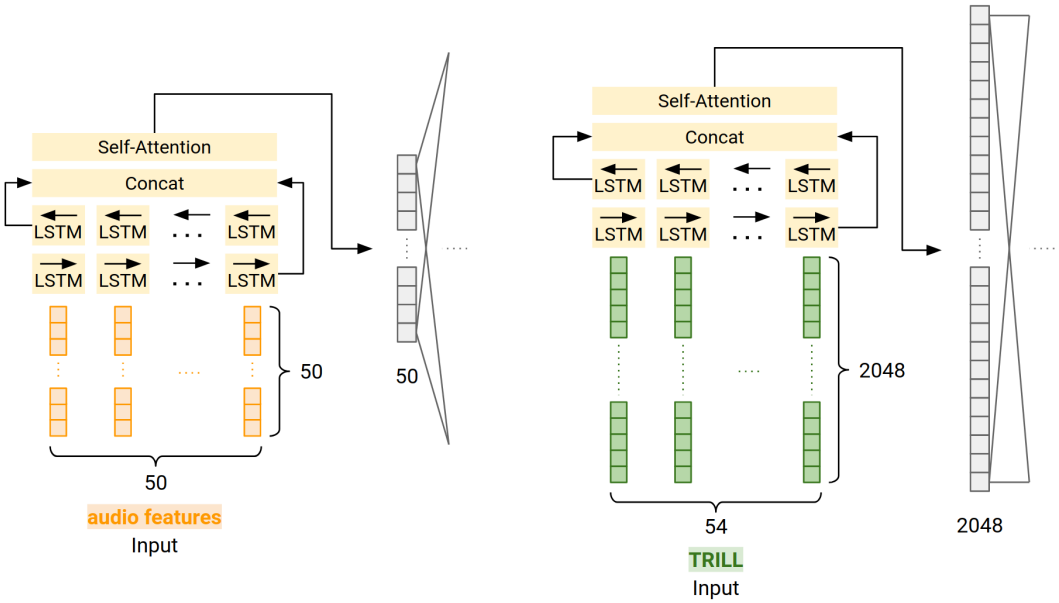

Figure 3.6: Architecture of the Bi-LSTM models. The following feed-forward neural networks are only being hinted at.

k-Nearest Neighbors

For k-Nearest Neighbors, the only variable we can influence is the choice of k. For each all three input features, we will be evaluating all values of k from 10 to 1000 in steps of 10. We will use the implementation provided by the *scikit-learn*[5] framework.

Random Forests

For our random forests, we will train 100 decision trees each and evaluate the performance of the maximum tree depth for depths between 1 and 20 in steps of 1. Again, we will use the implementation provided by *scikit-learn*.

	local machine	*server*
OS	Ubuntu 20.10	Ubuntu 18.04.5 LTS
CPU	AMD Ryzen 7 4700U @ 2.00GHz	Intel(R) Xeon(R) Silver 4114 @ 2.20GHz
RAM	16 GB	192 GB

Table 3.5: Parameters of the used hardware.

3.4.2 Hardware

The training and evaluation of the neural network based approaches was executed on the *ccblade5* server, which is part of the university infrastructure. kNN and random forests were trained and evaluated on our local machine. All models were trained on the CPUs. The parameters of both machines can be seen in Table 3.5.

3.4.3 Evaluation

Cross-Validation and Stratification

Section 2.3.5 introduced the concept of cross validation and stratification and concluded that 10-fold stratified cross validation should be used for model selection. We will follow this recommendation, but pay special attention to our stratification process.

We can not just randomly sample embeddings, but must either include or exclude all embeddings from a single speaker, as training data could otherwise 'leak' into the test set. We will therefore divide the sampling procedure into two steps:

1. Divide all speakers into 10 folds in a stratified manner
2. Sample features for the speakers in each fold

Our stratification process looks as follows: First, we sort all speakers by their likability score. Then, we divide them into 10 equal parts, such that the first part

[5] `https://scikit-learn.org/stable/`

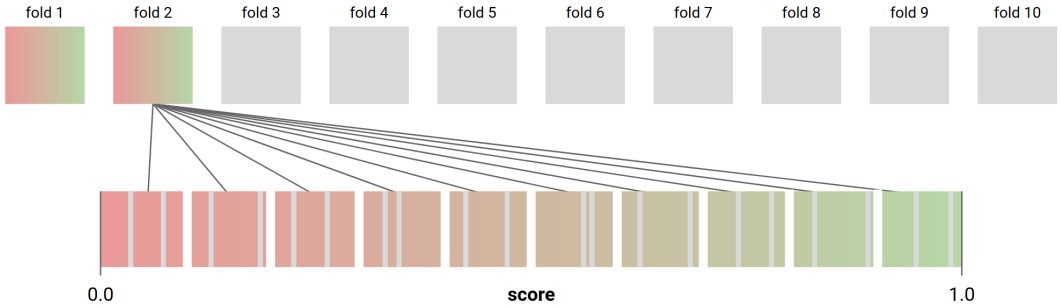

Figure 3.7: Illustration of the used stratification method.

contains the worst 10 percent of speakers and so on. For each of our folds, we then randomly draw 10 percent of speakers from each group. An illustration of this stratification method can be seen in Figure 3.7. This approach will give us some degree of randomness, while also making sure our folds fit the following criteria as much as possible:

- equal number of speakers per fold
- equal score average for speakers per fold
- equal score standard deviation for speakers per fold
- equal number of samples per fold

The composition of the resulting folds can be seen in Table 3.6. Finally, we are sampling the features for all folds by drawing embeddings or acoustic features for random audio samples for each speaker. How many samples are drawn per speaker depends on the number of articles spoken by them, but is capped at a maximum of 100 embeddings, so that very active speakers do not dominate single folds.

All methods and features will be evaluated using the exact same cross-validation and stratification technique, and all ten evaluation metrics will be recorded. We will report the average error across all folds, but also show the fold-wise error as box plots and use them to perform further statistical tests.

	fold									
	1	2	3	4	5	6	7	8	9	10
number of speakers	20	20	20	20	20	20	20	20	20	19
score avg. of speakers	0.480	0.480	0.485	0.480	0.478	0.478	0.485	0.487	0.483	0.451
score sd. of speakers	0.299	0.298	0.302	0.303	0.301	0.295	0.300	0.303	0.294	0.277
number of samples	514	511	450	489	530	480	640	506	500	440

Table 3.6: Amount, score average and score standard deviation of speakers and total number of samples per fold.

Mean Squared Error

Across all conditions, the **mean squared error** (MSE) [45] will be used as evaluation metric:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(score_{true} - score_{predicted})^2$$

To ensure comparability, we will also calculate the mean squared error for two baseline methods: random guessing, which will always guess a random score between 0 and 1, and mediocrity guessing, which will always guess a score of 0.5.

4 Results

After the preceding description of the experimental setup, this chapter will present the experiment results. For this purpose, the execution of the experiments is first summarized. Then the results obtained for each model are presented and, if applicable, compared with the results of previous work.

4.1 Experiment 1: Pairwise Classification

	features	*method*	*test set*			
			naïve	easy-0.25	easy-0.5	extern
reported in [19]	audio features + phonemes	siamese Bi-LSTM	0.67	0.93	0.97	—
	audio feautures	siamese Bi-LSTM	0.59	0.73	0.80	—
experimental	audio features + phonemes	siamese Bi-LSTM	0.67	0.81	0.90	**0.64**
	audio features	siamese Bi-LSTM	0.64	0.78	0.86	0.60
	ranks	rank-based	**0.72**	**0.82**	**0.95**	0.61

Table 4.1: Binary accuracy for classification of pairwise preferences. The best experimental result per test set is marked in boldface.

The training of both Bi-LSTM models took less than one hour each. In Table 4.1 the binary accuracies for the reproduced experiment and the new rank-based classifier are presented.

The results of the original study could only be partially reproduced, with the

model trained on audio features and phonemes performing considerably worse, and the model trained on audio features alone performing considerably better than reported. However, the inclusion of phonemes still resulted in better performance across all test sets. As expected, both models performed worse on the extern test set, in which one of the speakers has never been seen before. The creation of the extern test set resulted in a size of 140 samples.

The rank-based classifier consistently outperforms the Bi-LSTMs in the naïve, easy-0.25 and easy-0.5 test sets and performs slightly worse in the extern test set.

4.2 Experiment 2: Regression

Training and evaluation times for the models varied between only a few minutes for the k-nearest neighbors or random forest models and a maximum of 34 hours for the TRILL Bi-LSTM, with each fold taking 3.4 hours.

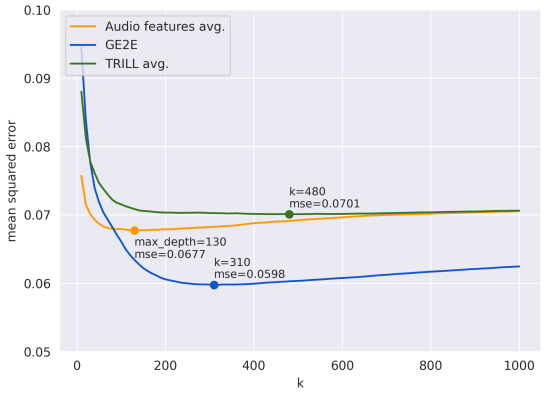

Figure 4.1: Mean squared error for different values of k using kNN regression. Lowest error value annotated.

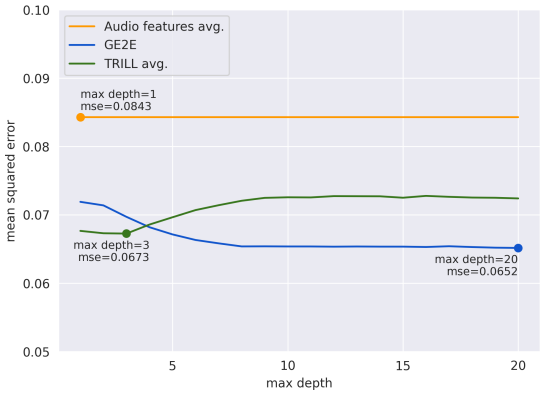

Figure 4.2: Mean squared error for different maximum tree depths using random forest regression. Lowest error value annotated.

The changes in mean squared error over k and the maximum tree depth can be seen in Figure 4.1 and 4.2. The optimal values of k are rather large with 130, 310 and 480 for audio features, GE2E and TRILL embeddings, which represent around 3, 7 and 11 percent of the samples in the training sets. The optimal values for

the maximum tree depth for random forests lie at 1, 20 and 3 in the same order. The mean square error stayed almost unchanged for all evaluated depths using the audio features, differing only in far off decimal places.

The best performing hyper-parameter combination for the feed-forward deep neural networks was a learning rate of 5e-8 and a dropout rate of 0.5, which were consequently used as parameters for the Bi-LSTM. The largest improvement could be observed for the increase of the dropout rate. Early stopping typically terminated the learning process after 20 to 40 epochs.

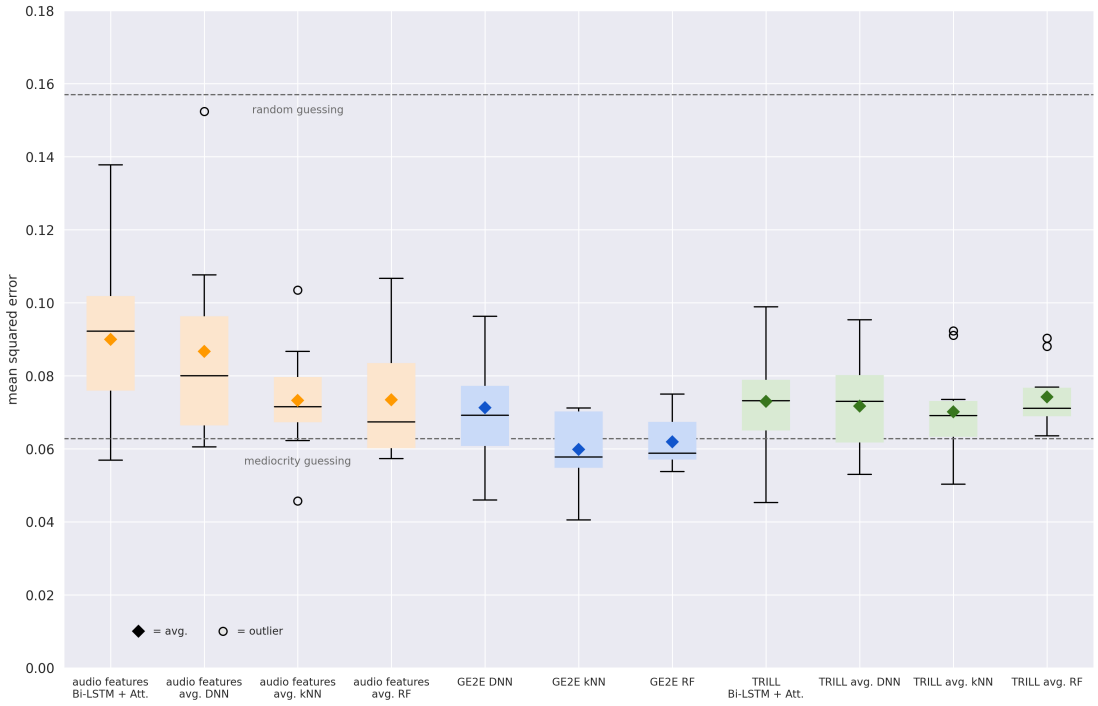

Figure 4.3: Boxplots of the mean squared error for all 10 folds per feature type and learning method for regression of likability scores. Baseline performances are shown as dashed lines.

Per condition, the fold-wise mean squared error are reported as box plots in Figure 4.3. The mean squared errors of the random guessing and mediocrity baseline are 0.016 and 0.063 respectively. Additionally, we report the average mean squared error across folds in Table 4.2 for quantitative comparison. Across all feature types, the k-nearest neighbor regressor performed best, followed by

random forests, feed-forward deep neural networks and Bi-LSTMs in that order. When looking at each learning method individually, GE2E embeddings performed best, followed by TRILL embeddings and audio features. These two observations hold without exception. No fold in any condition performed worse than the random guessing baseline. A lower average mean squared error than in the mediocrity baseline can only be observed for the GE2E k-nearest neighbor ($p = .20$) and random forest regressor ($p = .36$). Overall, the best combination was a k-nearest neighbor regressor on GE2E embeddings with an average mean squared error of 0.060.

		learning method			
		Bi-LSTM + attention	DNN	kNN	RF
features	audio features	0.090	0.086	0.073	0.073
	GE2E embeddings	—	0.071	**0.060**	0.062
	TRILL embeddings	0.073	0.072	0.070	0.074

Table 4.2: Average mean squared error across folds for regression of likability scores. The overall best result is marked in boldface.

Besides the mean squared errors, all individual predictions of the models were recorded per fold. This allows us to determine how often each model has predicted different scores. For this purpose, we have created density plots (see Figure 4.4) that reflect how well the model is able to reproduce the uniform distribution over scores in the training data. Not one of the models was able to do this, and the predictions of some of the models (e.g. TRILL Bi-LSTM) are so narrow that they only span over 10 percent of the available spectrum of scores in a single fold. Across input features, the models trained on TRILL embeddings were the most prone to this phenomenon. Across learning methods the differences are more nuanced, but the k-nearest neighbor and random forest regressors are the least susceptible and spread out their predictions the most. Nonetheless, predicted scores over 0.75 and under 0.25 are almost never seen here as well (less than one percent of all predictions), even though they make up 50 percent of all data.

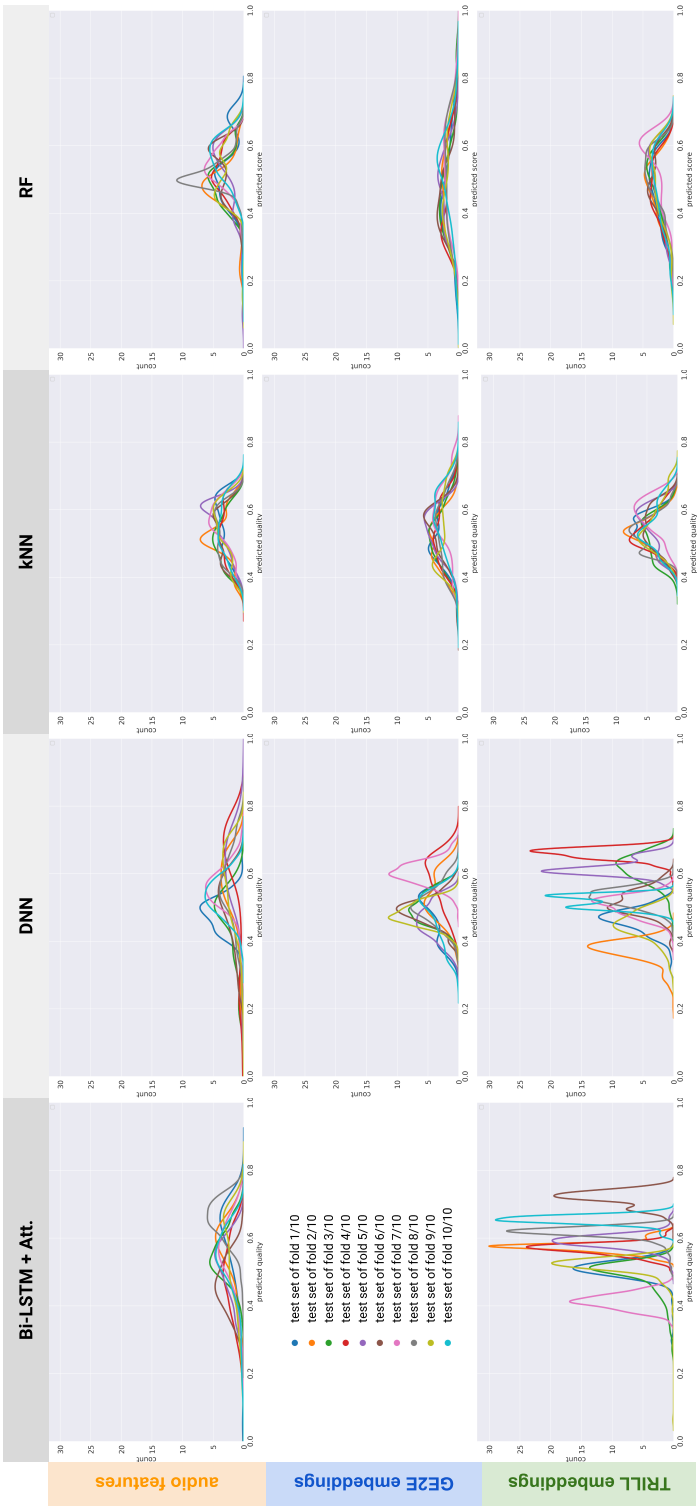

Figure 4.4: Density functions of predicted scores for all conditions.

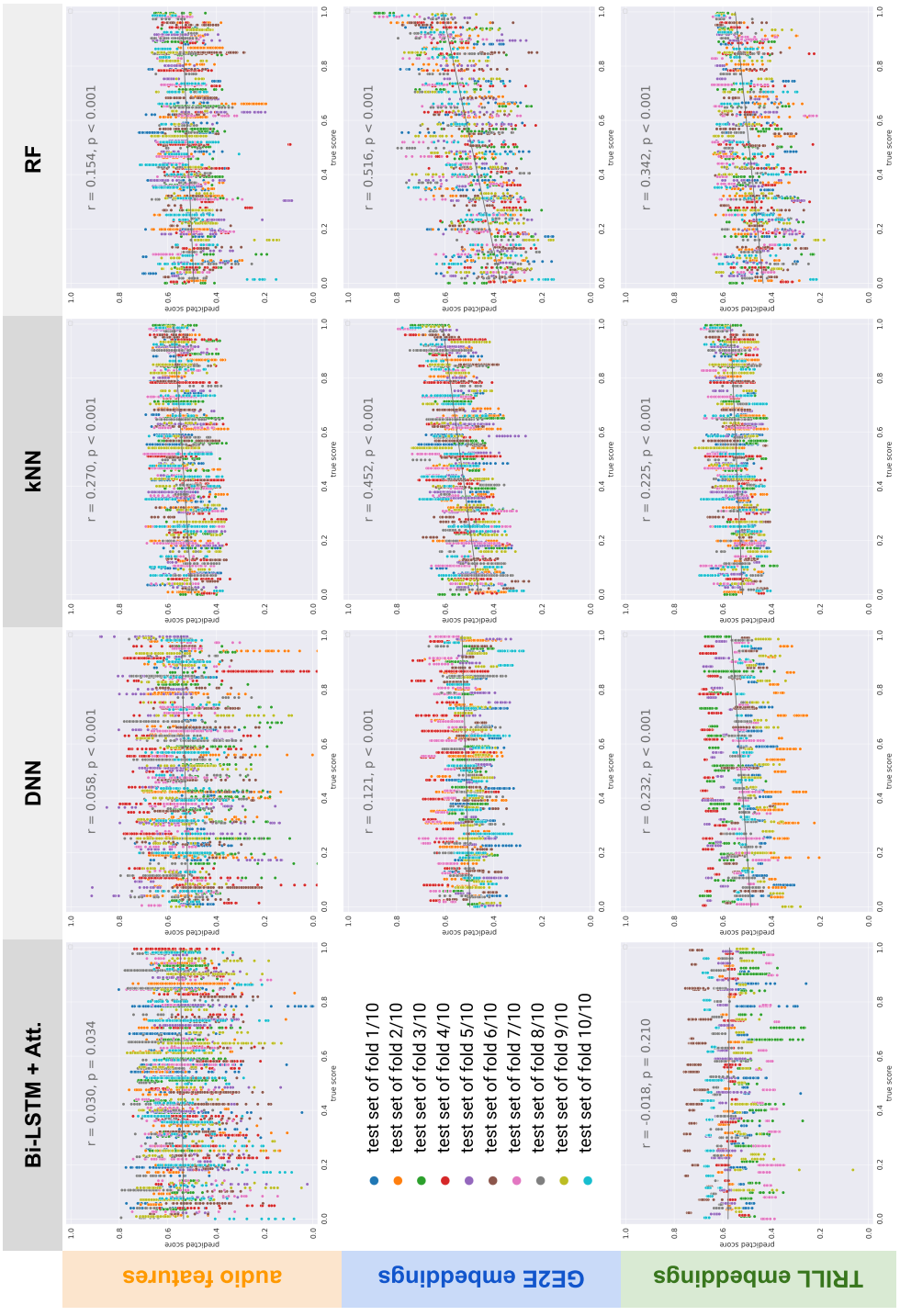

Figure 4.5: Scatter plots with true vs predicted score for all conditions. Vertical 'lines' represent multiple embeddings with the same true score and therefore the same speaker.

We are also reporting scatter plots (see Figure 4.5), in which true and predicted score are plotted per sample and fold. This allows us to visually inspect patterns formed by the predictions. Furthermore, we report Pearson's r as measure of linear correlation and visualize the slope. We are able to notice a number of properties:

1. Almost all predicted values fall in the range between 0.25 and 0.75. This is not surprising, as we were able to observe the same in the density plots in Figure 4.4.
2. There are clearly visible vertical 'lines'. These lines are formed by many dots with the same likability score and belong to the same speaker. How far one such line is spread out over the predicted scores tells us how consistent the model is with predictions for the same person. However, one should keep in mind that up to 100 dots exist per speaker, many of which are on top of each other. If the dispersion is normally distributed, it will visually appear larger than it actually is.
3. For most models only a weak positive correlation exists between true and predicted scores (if one exists at all). Almost all correlations are significant ($p < 0.001$). Only two models have a moderate positive correlation, which are the kNN and random forest regressor, both trained on GE2E embeddings, with a correlation of 0.452 and 0.516 respectively.

Another way of looking at the model consistency is viewing the intra-speaker score distribution as measure for the text-independence of the model, one of the key features we were trying to achieve. In order to be able to make quantitative statements, we are additionally reporting the average intra-speaker spread (difference between highest and lowest predicted score per speaker) as well as the average intra-speaker standard deviation (standard deviation of all predicted scores per speaker) in Table 4.3.

Generally, the intra-speaker variability seems to be the lowest for all methods trained on TRILL embeddings. The highest observed standard deviation is 0.083 for both neural-network-based approaches trained on audio features with a spread of 0.31. The lowest value for both span and standard deviation can be observed for the Bi-LSTM trained on TRILL embeddings, with an average maximum span of 0.1 and an average standard deviation of just 0.028. However, it is important to

features			*learning method* Bi-LSTM + attention	DNN	kNN	RF
	audio features	*span*	0.31	0.31	0.16	0.18
		sd	0.083	0.083	0.044	0.050
	GE2E embeddings	*span*	—	0.12	0.15	0.23
		sd	—	0.033	0.041	0.073
	TRILL embeddings	*span*	**0.10**	0.10	0.11	0.16
		sd	**0.028**	0.031	0.031	0.053

Table 4.3: Average intra-speaker span between highest and lowest predicted score and average intra-speaker standard deviation of predicted scores. Overall lowest scores marked in boldface.

look at where the text-independence is coming from. If we remind ourselves of the corresponding density plot, we see that the reason for the low standard deviation is the extreme overfitting to individual folds.

While a low variability per speaker speaks for excellent text-independence, these results have to be interpreted very carefully. Always guessing 0.5 will yield a span and standard deviation of 0, and is—per definition—fully text-independent. However, we want both: a low intra-speaker standard deviation and a high inter-speaker standard deviation. To quantify this, we are dividing the average intra-speaker standard deviation per fold by the standard deviation of all predicted scores in the respective fold and reporting the average of the 10 resulting proportions for each condition in Table 4.4. We will refer to this metric as *intra-inter-ratio*.

If the calculated value is high, a higher proportion of the average intra-speaker standard deviation can be explained by the intra-fold standard deviation. If the value is low, scores per speaker are distinct, but diversely distributed across speakers in each fold. All values will lie between 0 and 1 and represent the percentage of explained dispersion. With this metric, the previous champions are now taking a back seat. The new winner in terms of *meaningful* text-independence is now

		learning method			
		Bi-LSTM + attention	DNN	kNN	RF
features	audio features	0.723	0.663	0.549	0.536
	GE2E embeddings	—	0.483	**0.393**	0.480
	TRILL embeddings	0.536	0.480	0.468	0.442

Table 4.4: Intra-inter-ratio across all conditions. Lower scores are better. Overall lowest scores marked in boldface.

the kNN regressor trained on GE2E embeddings, where only 39 percent of the standard deviation originate from the overall noise.

This is not a coincidence. If we compare the calculated ratios with the mean squared error of each approach, as seen in the scatter plot in Figure 4.6, we will find that both of them go hand in hand. The reason is that our desire for text-independence was hard-coded in all of our models. The mere fact that we gave each speaker the same score across all samples was enough to optimize for text-independence, no matter which input features we selected.

As we can see, the k-nearest neighbor regressor has both the lowest mean squared error as well as the highest meaningful text-independence. It is closely followed by the random forest regressor trained on the same embeddings.

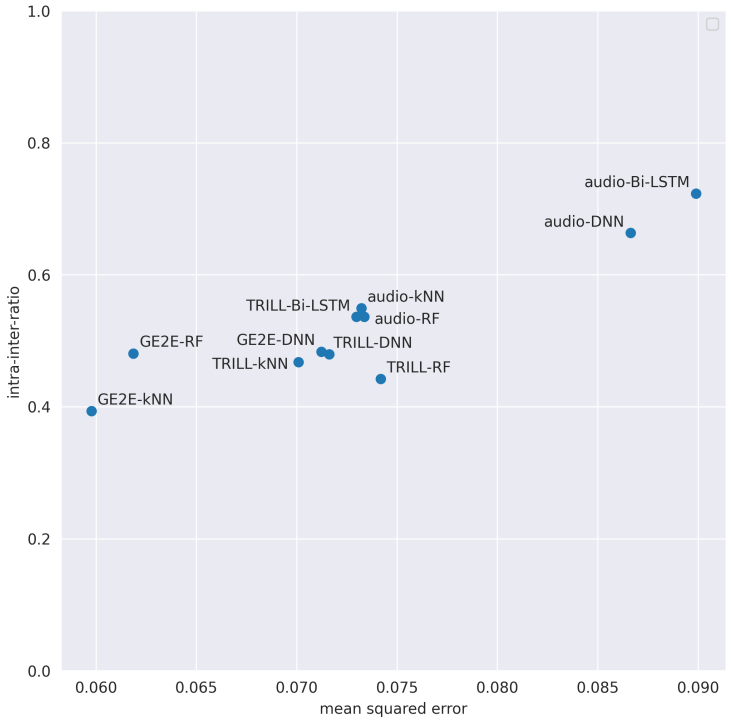

Figure 4.6: Average mean squared error plotted against the intra-inter-ratio for each condition.

5 Discussion

After the reporting of the experimental results and a selection of derived metrics, we will now discuss the outcomes. Whenever helpful for our further assessment, we will perform minor validation tests to substantiate our findings.

5.1 Experiment 1: Pairwise Classification

The results of the original study could be partially reproduced. While a clear advantage of the inclusion of phonemes was evident, the performance of the audio features was better and the performance of the combined audio features and phonemes worse than expected.

The precise numbers of our results should definitely be taken with a grain of salt, as the three original test sets consisted only of 100 speakers each, and a difference in accuracy of for example 7 percent in the phoneme-easy-0.5 condition corresponds to only 7 differently classified samples. It is quite likely that another run will yield slightly different results. However, this should not be taken as evidence in favor of the reported accuracies, but rather a set of hypothetical accuracies somewhere between both.

Especially since no additional features on the stimuli or raters were used, the meaningfulness of the results is greatly diminished. If the aim was not to learn how individual preferences of raters or meta data of the speakers influence the rating decision, the only thing that can be learned is a noisy representation of ranks.

This is also our main criticism: A performance better than the rank-based classifier can never be expected—not with any learning method—if identical inputs are presented with contradicting outputs. The fact that stimuli are able to show

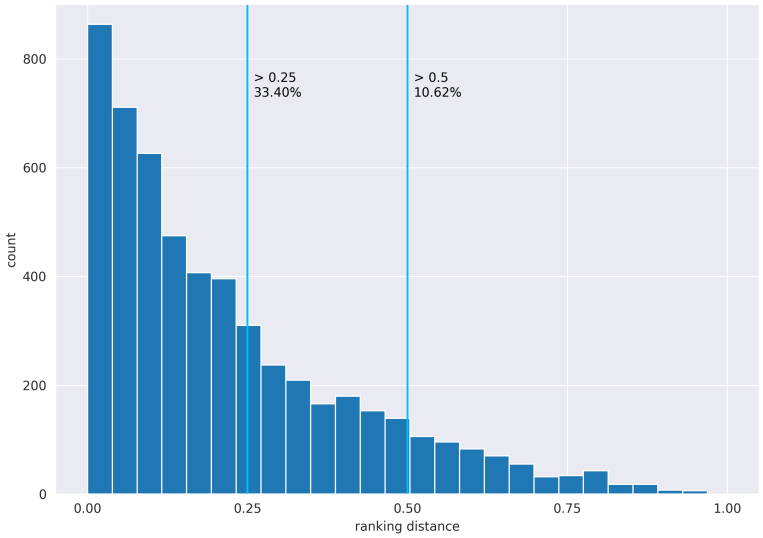

Figure 5.1: Histogram of pairwise rating distances with cumulative proportions for a distance larger than 0.5 and 0.25 using active sampling.

up both in the train and test set degrades them to mere identifiers, which in the best case are at some point associated with their respective rank and in the worst case add additional unwanted noise.

It could therefore be expected that the rank-based classifier outperformed the Bi-LSTMs by a margin in all three original test sets, and it is unlikely that more runs will yield a significantly different result. Additional fine-tuning could improve the results, but surpassing the rank-based baseline is practically impossible. The evaluation of different and non-contradictory features will likely be a more reasonable choice.

The reported high accuracies are looking very pleasing at first sight, but represent only minor portions of the available data. The active sampling process, which strongly favored 'competitive' comparisons, leads to only a third of the samples being more than 0.25 apart, and only 11 percent of samples have a distance larger than 0.5, as can be seen in Figure 5.1.

On a more positive note, we want to point out that the combination of audio

features and phonemes did beat the rank-based approach in the extern test set. This is the only test set where the features could not just be used to memorize ranks, but also predict ranks, even if only in an indirect manner. For us, the switch from underperforming to overperforming the baseline is certainly worth noting, and further investigations into phoneme-specific speech quality models could easily be worth the cost.

Overall, the study was able to show that the use of phoneme alignments is able to improve accuracy for pairwise classification of likability ratings, and we expect this finding to hold for different paralinguistic tasks as well. On a further note, foreign accents seem to have had a considerable effect on the ratings (for example, the worst ranked speaker has a very pronounced eastern-european accent). However, the SWC corpus is too small and homogeneous to explore this in a systematic manner, and it would probably be easier to create new likability scores on a corpus with a larger dialect-related and/or accent-related diversity.

Finally, the publication served not just as subject, but also as inspiration for this thesis, which we consider noteworthy as well. Compared to computer vision, the area of speech processing is still in its beginnings in terms of the wide-spread adoption of machine learning models, and we hope the author will continue to contribute to this field.

5.2 Experiment 2: Regression

The audio features were only able to deliver satisfactory results in the pairwise classification task. The problem did however not lie in the features themselves, but the structure of the experiment setup, and especially the composition of the test sets. The single-sample regression approach allows the inputs to take full credit for the achieved scores, without dilution by implicit ranking advantages. It is therefore a more neutral way of evaluation, at least for this specific task.

Overall, the results were underwhelming. No method was able to achieve a mean squared error that was significantly below our mediocrity baseline, which was not what we expected. However, we must not only consider, *what* performance was achieved, but also *how* this performance was achieved. A method that 'cheats'

by always predicting a score of 0.5 (or any other value), can be easily identified by a prediction score standard deviation of zero. If the standard deviation is not zero, but equally good performance were achieved, the model must have learned another way of getting there.

We can compare the situation to a student who was able to score a very good grade in an exam. If they did not cheat, and we can even *verify* this by some measurement, the student must have either a) been lucky or b) learned something. All the additional bells and whistles we are using, be it cross-validation or statistical tests, ultimately have the goal to provide so much evidence against a) that we can in good conscience accept b) as fact.

5.2.1 Discussion of Methods

Several mistakes have been made in the planning of the experiments, which can decrease our confidence in the b) for our models. They are as follows:

- Insufficient separation of training and testing data
- Uneven selection of samples across speakers, despite stratification
- Misrepresentative modeling of the underlying population
- Incautious selection of error function

We will look at each mistake one after the other to determine if harm was done, and if so, how that must influence our conclusions.

Insufficient Separation of Training and Testing Data

Across all conditions, we performed hyper-parameter optimization in the outer loop. This means all hyper-parameters were evaluated using 10-fold stratified cross-validation without the use of a dev set and then compared. This method is less reliable (for reporting model performance) than searching for hyper-parameters inside of train-dev-sets. However, it is also hard to imagine that this distorted or results so much that we would have come to other conclusions.

The bigger mistake, which did have observable effects, is the application of early stopping on the test sets. The training of all neural networks, deep feed-forward and Bi-LSTM, was stopped when the mean squared error on the corresponding test set stopped decreasing. This repeated 'peeking' into the test set is a no-go and resulted in the partially very narrow prediction score corridors visible in the density plots (Figure 4.4). We are glad that the effect has not been more pronounced on audio features and GE2E embeddings, where the predictions are still spread out reasonably well.

This mistake is highly problematic, as it can artificially decrease our measured model error. Luckily, we are off the hook: The neural network models did not perform exceptionally well anyway. If they could not reach a good performance with cheating, they will also not reach it without. This mistake will not change our overall findings, even if just out of pure luck. We therefore consider this mistake **not harmful**.

Uneven selection of samples across speakers

A lot of attention has been given to the fold stratification process. Speakers were selected according to their rank, and speaker means and standard deviations were aligned. However, as the number of selected samples was allowed to vary between 10 and 100, the equal means could not be preserved when looking at all sample scores in one fold at once. Luckily, the baselines were calculated on the exact same folds, and thus are reliable. However, if one wanted to train an unbiased model, much more care would have to be taken with respect to the distributions across all samples, but also parameters such as age and sex. Since the folds were identical across all conditions, including the baselines, and differed only slightly, we consider this mistake **not harmful**.

Misrepresentative population modeling

Likability ranks were transformed linearly onto the $[0, 1]$ score interval. However, likability scores, when assessed in increments, e.g. using a Likert-like scale, have been found to be roughly normally distributed and quite symmetric [12]. As

disagreement was more frequently observed for speakers with ranks in the middle, which can be seen in Figure 3.2, it comes quite naturally that our data must have reflected a normally distributed variable as well.

This simple difference has an enormous effect, as speakers that are equally 'close' on a normally distributed scale can be either pulled apart in the middle or squished together at the borders if forced into a uniform distribution. These different (yet equal) distances find their way into our evaluation metric, the mean squared error. We consider this mistake to be **harmful**, as our models are equipped with different abilities to cope with non-linearity and are therefore not comparable any more.

Incautious error function selection

The mean squared error has been selected as performance metric because it is one of the most used error functions. However, we should very thoroughly think about what we want to measure and optimize. The mean squared error has the assumption built in that one mistake of size x is much worse than two mistakes of half the size. If this assumption holds for the prediction of likability scores is questionable. The very visible side-effect of our error function is that all models were severely punished for predictions close to the borders of the score interval at 0 and 1, as they will, on average, have higher distances between true and predicted score. Other error functions, such as the mean absolute error, will of course also punish larger distances more, but not as severely as the mse.

We consider this mistake to be **not harmful**, even though it likely resulted in much narrower predicted scores than necessary and therefore had a clearly visible qualitative effect on our models. However, as the aim of this thesis was not the development of a good model, but a reliable comparison across conditions and baselines, we think this can be mainly neglected.

5.2.2 Further Testing

If the classifications into harmful and not harmful made above hold in every circumstance can certainly be debated. Therefore, we will provide further evidence

features	*learning method* kNN	RF
audio features	0.57	0.58
GE2E embeddings	0.64	0.66
TRILL embeddings	0.59	0.61

Table 5.1: Binary accuracies for per-sample classification.

to substantiate our findings. We will evaluate our best performing models under significantly simplified circumstances.

We will evenly divide the speakers into two classes, containing the top and bottom 50 percent in the ranking. Then, we will turn our most promising model, the GE2E-kNN regressor, into a GE2E-kNN classifier. We will then report the performance of the classifier and a random guessing baseline. This setup eliminates the influence of most mistakes:

- Both normally and uniformly distributed scores will be split up in the middle, resulting in identical binary classes due to their symmetry.
- The mean squared error will be replaced by binary accuracy as error function. As both classes are balanced, we will not report recall.
- Additionally, to err on the side of caution, we will take an equal amount of exactly 10 samples for all speakers this time.
- We will keep the 10-fold stratified cross-validation, but not perform any additional hyper-parameter search. Values for k will be adjusted downwards by , as the data set has shrunken in size. Depth values for random forests will stay the same.

Because this type of evaluation is computationally comparatively cheap, we will also build a random forest classifier and let both models run on the TRILL embeddings and audio features as well. The resulting binary accuracies can be seen in Table 5.1. The baseline accuracy for random guessing has been recorded as 0.48.

It can be seen that GE2E embeddings are again the top performers, with an average binary accuracy of 0.64 and 0.66 for the kNN and random forest classifier. All six accuracies are significantly higher than the baseline ($p < 0.001$). Even though the differences across features are a lot smaller this time, GE2E embeddings still perform significantly better than audio features using the kNN ($p < 0.05$) and random forest ($p < 0.01$) models.

5.2.3 Ethical Considerations

Greatly simplified, bias in a machine learning model can come from three sources:

1. The machine learning model does not reflect the data.
2. The data does not reflect how the world is.
3. The world does not reflect how we want the world to be.

In our case, we already know that the first source of bias has a huge influence on the output of our models. Even if excellent or terrible speakers are presented, the model fails to give them a proper score, and rather returns values between 0.25 and 0.75. Many more oddities have been reported in the previous section. This kind of statistical bias is generally well understood, and remedies against it are known.

We also know that the training data we are using does not reflect how the world is. This begins with the discrepancy between uniformly and normally distributed scores, but reaches to just having 20 females among the ranked speakers and the raters being overwhelmingly male German computer science students. This is certainly problematic, as Baumann reported that 'only a moderate correlation [...] between female and male listener rankings' exists, indicating 'different preferences between these listener groups' [20]. Equally, the 'rank assigned to a female speaker is on average 12.7 ranks better for female than for male listeners' [20]. The biases reported in this paragraph are only touching on the topic, while leaving out many more aspects such as age-related or origin-related bias.

Biases in the second category must not even stem from our own choice of data collection, but can be passed on via transfer learning. In our case, some bias could

already originate in the data used to train the embeddings. The speakers in the LibriSpeech and VoxCeleb data sets are certainly not a truthful representation of the average person on earth, and this does not just hold for age and language, but also socio-economic factors and the related sociolects.

The last source of bias is one that we can not objectively measure. Yet, it is hard to imagine a world with no room for improvement.

While we as researchers are able to assess different aspects of bias in an academic setting, most people who are eventually rated by machine learning systems do not have these possibilities. Furthermore, the persons deploying machine learning models are quite often not identical to the ones who developed them. This makes it even more important to document and communicate the scope and limitations of each model in a way that can easily be understood and prevent the application of models to use-cases for which they were not built.

Our models were constructed for testing an hypothesis and to serve as a proof-of-concept. They are not meant to be deployed in any real-world application scenario, and certainly not intended to make reliable judgments about specific individuals.

5.2.4 Future Work

Future work in this area should mainly look at the performance of the evaluated methods across other—potentially more diverse—data sets. Especially the availability of only one reliably rated speech sample per speaker made it hard to make statements about how well specific speakers really perform, and how much perceived likability is able to change over the course of one article.

Alternatively, the suitability of more kinds of embeddings could be evaluated, for example *Unspeech* [72] embeddings. Additionally, embeddings could tried to be learned from scratch, or fine-tuned based on a pre-trained model, instead of just applied.

The likable-trait hypothesis we established in the introduction was called into question by the discovery of identical speakers that got assigned ranks that are very far apart, and a more detailed investigation into this phenomenon would

likely yield interesting results. If changes in perceived likability over time could be collected on a large scale and later predicted from the speech signal, speakers could be given feedback that helps them improve their quality. One is not able to replace ones innate voice, but certainly to change pronunciation or rhythm. Especially the observed role of phonemes appears to be of interest, but will also be highly specific to the spoken language.

Another approach could be a further evaluation not just of how, but also why specific embeddings correspond with a high likability score. One could imagine removing different subsets of dimensions from the embedding and recording the influence this has on predicted likabilities. Alternatively, correlates between score and dimension could be calculated to identify dimensions that have a strong influence on the predicted score. Such dimensions will likely also have high correlates with the sex of the speaker on the data set used in our experiments.

Focus could also be put not just on the prediction, but the generation and synthesis of likable speech. Visible progress is currently being made in this field, even though good results will likely require data sets that are larger by orders of magnitude.

6 Conclusion

This thesis has investigated whether speech embeddings can be used as text-independent input features for the prediction of speech likability scores.

First, previous research on the binary classification of pairwise likability ratings using audio features and phonemes was examined. The experiments were only partially reproducible, as one model showed slightly worse and one model slightly better performance than expected. When compared to a newly developed baseline, both models revealed weaknesses in their conception.

Next, traditional audio features, GE2E embeddings and TRILL embeddings were used to train Bi-LSTMs with attention, deep feed-forward neural networks, k-nearest neighbor regressors and random forest regressors. All possible combinations were evaluated in a systematic manner, in addition to two baseline methods. The results were then analyzed in great depth, with special attention on the interplay between performance and text-independence.

Both in terms of a low mean squared error as well as a high and meaningful speech-independence, the combination of GE2E embeddings and a k-nearest neighbor regressor outperformed all other conditions. Even though the mean squared error of this approach was not significantly different from a simple baseline method, further analysis showed that the approach used for the baseline was not copied by the regressor. This model is therefore—to a certain degree—able to derive the correct likability score from information entailed in the embeddings.

After an identification of potential shortcomings of the experiment, the performance of the three different input features was evaluated again, this time for sample-wise binary classification into good and bad speakers using a k-nearest neighbor and random forest classifier. All combinations delivered binary accuracies

that were significantly better than chance. This time, the combination of GE2E embeddings and random forests performed best.

Finally, we laid out some ethical concerns about bias in the model and provided directions for further research.

Bibliography

[1] N. Lass, K. Hughes, M. Bowyer, L. Waters and V. Bourne, 'Speaker sex identification from voiced, whispered, and filtered isolated vowels', *The Journal of the Acoustical Society of America*, vol. 59, no. 3, pp. 675–678, Mar. 1976, ISSN: 0001-4966. DOI: 10.1121/1.380917. [Online]. Available: https://doi.org/10.1121/1.380917 (cit. on p. 2).

[2] S. M. Hughes and B. C. Rhodes, 'Making age assessments based on voice: The impact of the reproductive viability of the speaker', *Journal of Social, Evolutionary, and Cultural Psychology*, vol. 4, no. 4, pp. 290–304, 2010, ISSN: 1933-5377(Electronic). DOI: 10.1037/h0099282. [Online]. Available: https://doi.org/10.1037/h0099282 (cit. on p. 2).

[3] J. H. Walton and R. Orlikoff, 'Speaker race identification from acoustic cues in the vocal signal', *Journal of speech and hearing research*, vol. 37 4, pp. 738–45, 1994 (cit. on p. 2).

[4] N. J. Lass and M. Davis, 'An investigation of speaker height and weight identification', *The Journal of the Acoustical Society of America*, vol. 60, no. 3, pp. 700–703, Sep. 1976, ISSN: 0001-4966. DOI: 10.1121/1.381142. [Online]. Available: https://doi.org/10.1121/1.381142 (cit. on p. 2).

[5] B. L. Brown and W. E. Lambert, 'A cross-cultural study of social status markers in speech', *Canadian Journal of Behavioural Science / Revue canadienne des sciences du comportement*, vol. 8, no. 1, pp. 39–55, 1976. DOI: 10.1037/h0081933. [Online]. Available: https://doi.org/10.1037/h0081933 (cit. on p. 2).

[6] K. R. Scherer, R. Banse and H. G. Wallbott, 'Emotion inferences from vocal expression correlate across languages and cultures', *Journal of Cross-cultural psychology*, vol. 32, no. 1, pp. 76–92, 2001 (cit. on p. 2).

[7] J.-J. Aucouturier, P. Johansson, L. Hall, R. Segnini, L. Mercadié and K. Watanabe, 'Covert digital manipulation of vocal emotion alter speakers' emotional states in a congruent direction', *Proceedings of the National Academy of Sciences*, vol. 113, no. 4, pp. 948–953, 2016, ISSN: 0027-8424. DOI: 10.1073/pnas.1506552113. eprint: https://www.pnas.org/content/113/4/948.full.pdf. [Online]. Available: https://www.pnas.org/content/113/4/948 (cit. on p. 2).

[8] R. M. Krauss, R. Freyberg and E. Morsella, 'Inferring speakers' physical attributes from their voices.', *Journal of Experimental Social Psychology*, vol. 38, no. 6, pp. 618–625, 2002. DOI: 10.1016/S0022-1031(02)00510-3. [Online]. Available: https://doi.org/10.1016/S0022-1031(02)00510-3 (cit. on p. 2).

[9] T.-H. Oh, T. Dekel, C. Kim, I. Mosseri, W. T. Freeman, M. Rubinstein and W. Matusik, *Speech2Face: Learning the face behind a voice*, 2019. arXiv: 1905.09773 [cs.CV] (cit. on p. 2).

[10] F. Eyben, M. Wöllmer and B. W. Schuller, 'Opensmile: The munich versatile and fast open-source audio feature extractor', in *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, A. D. Bimbo, S. Chang and A. W. M. Smeulders, Eds., ACM, 2010, pp. 1459–1462. DOI: 10.1145/1873951.1874246. [Online]. Available: https://doi.org/10.1145/1873951.1874246 (cit. on p. 2).

[11] J. Pohjalainen, S. Kadioglu and O. Räsänen, 'Feature selection for speaker traits', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 270–273. [Online]. Available: http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0270.html (cit. on p. 2).

[12] B. W. Schuller, S. Steidl, A. Batliner, E. Nöth, A. Vinciarelli, F. Burkhardt, R. van Son, F. Weninger, F. Eyben, T. Bocklet, G. Mohammadi and B. Weiss, 'A survey on perceived speaker traits: Personality, likability, pathology, and the first challenge', *Comput. Speech Lang.*, vol. 29, no. 1, pp. 100–131, 2015. DOI: 10.1016/j.csl.2014.08.003. [Online]. Available: https://doi.org/10.1016/j.csl.2014.08.003 (cit. on pp. 2, 3, 11, 58).

[13] C. Montacié and M. Caraty, 'Pitch and intonation contribution to speakers' traits classification', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 526–529. [Online]. Available: `http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0526.html` (cit. on p. 2).

[14] L. P. Coelho, D. Braga, M. S. Dias and C. Garcıa-Mateo, 'An automatic voice pleasantness classification system based on prosodic and acoustic patterns of voice preference', in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, ISCA, 2011, pp. 2457–2460. [Online]. Available: `http://www.isca-speech.org/archive/interspeech%5C_2011/i11%5C_2457.html` (cit. on p. 2).

[15] D. Wu, 'Genetic algorithm based feature selection for speaker trait classification', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 294–297. [Online]. Available: `http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0294.html` (cit. on p. 2).

[16] H. Buisman and E. O. Postma, 'The log-gabor method: Speech classification using spectrogram image analysis', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 518–521. [Online]. Available: `http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0518.html` (cit. on p. 2).

[17] F. Burkhardt, B. Schuller, B. Weiss and F. Weninger, 'Would you buy a car from me? - On the likability of telephone voices', Jan. 2011, pp. 1557–1560 (cit. on pp. 2, 11).

[18] R. Brueckner and B. W. Schuller, 'Likability classification - A not so deep neural network approach', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 290–293. [Online]. Available:

http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0290.html (cit. on p. 2).

[19] T. Baumann, 'Learning to determine who is the better speaker', in *Proc. 9th International Conference on Speech Prosody 2018*, 2018, pp. 819–822. DOI: 10.21437/SpeechProsody.2018-165. [Online]. Available: http://dx.doi.org/10.21437/SpeechProsody.2018-165 (cit. on pp. 2, 12, 16, 17, 25, 27, 33, 34, 36, 44).

[20] ——, 'Large-scale speaker ranking from crowdsourced pairwise listener ratings', in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed., ISCA, 2017, pp. 2262–2266. [Online]. Available: http://www.isca-speech.org/archive/Interspeech%5C_2017/abstracts/1697.html (cit. on pp. 2, 12, 16, 27–30, 34–36, 61).

[21] T. Baumann, A. Köhn and F. Hennig, 'The spoken wikipedia corpus collection: Harvesting, alignment and an application to hyperlistening', *Lang. Resour. Evaluation*, vol. 53, no. 2, pp. 303–329, 2019. DOI: 10.1007/s10579-017-9410-y. [Online]. Available: https://doi.org/10.1007/s10579-017-9410-y (cit. on pp. 2, 26).

[22] L. Wan, Q. Wang, A. Papir and I. Lopez-Moreno, 'Generalized end-to-end loss for speaker verification', in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, IEEE, 2018, pp. 4879–4883. DOI: 10.1109/ICASSP.2018.8462665. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8462665 (cit. on pp. 3, 22, 23).

[23] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. de Chaumont Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel and Y. Haviv, 'Towards learning a universal non-semantic representation of speech', in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu and T. F. Zheng, Eds., ISCA, 2020, pp. 140–144. DOI: 10.21437/Interspeech.2020-1242. [Online]. Available: https://doi.org/10.21437/Interspeech.2020-1242 (cit. on pp. 4, 22–24).

[24] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing (4. ed.)* Prentice Hall, 2006, ISBN: 0131873741 (cit. on pp. 6–8).

[25] X. Huang, A. Acero, H.-W. Hon and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st. USA: Prentice Hall PTR, 2001, ISBN: 0130226165 (cit. on pp. 7, 8).

[26] S. Wood, 'Non-negative matrix decomposition approaches to frequency domain analysis of music audio signals', PhD thesis, Université de Montréal, Apr. 2010 (cit. on p. 7).

[27] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding And Error Concealment*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006, ISBN: 0470031743 (cit. on pp. 8, 9).

[28] M. Ajili, 'Reliability of voice comparison for forensic applications', PhD thesis, Université d'Avignon, Nov. 2017. DOI: 10.13140/RG.2.2.15627.13605 (cit. on p. 8).

[29] G. L. Trager, 'Paralanguage: A first approximation', *Studies in Linguistics*, vol. 13, pp. 1–12, 1958 (cit. on pp. 9, 10).

[30] *Paralanguage*, https://dictionary.cambridge.org/dictionary/english/paralanguage, Accessed: 2021-04-18 (cit. on p. 9).

[31] *Paralanguage*, https://www.merriam-webster.com/dictionary/paralanguage, Accessed: 2021-04-18 (cit. on p. 10).

[32] S. Johar, *Emotion, Affect and Personality in Speech: The Bias of Language and Paralanguage*, 1st. Springer Publishing Company, Incorporated, 2015, ISBN: 3319280457 (cit. on p. 10).

[33] B. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Oct. 2013, pp. 1–321, ISBN: 9781119971368. DOI: 10.1002/9781118706664 (cit. on pp. 10, 11).

[34] ——, *Interspeech Computational Paralinguistics ChallengE (ComParE)*, http://www.compare.openaudio.eu/, Accessed: 2021-04-18 (cit. on p. 11).

[35] L. F. Gallardo, G. Mittag, S. Möller and J. Beerends, 'Variable voice likability affecting subjective speech quality assessments', in *Tenth International Conference on Quality of Multimedia Experience, QoMEX 2018, Cagliari, Italy, May 29 - June 1, 2018*, IEEE, 2018, pp. 1–6. DOI: 10.1109/QoMEX.2018.8463432. [Online]. Available: https://doi.org/10.1109/QoMEX.2018.8463432 (cit. on p. 11).

[36] B. Weiss and F. Burkhardt, 'Voice attributes affecting likability perception', in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, T. Kobayashi, K. Hirose and S. Nakamura, Eds., ISCA, 2010, pp. 2014–2017. [Online]. Available: http://www.isca-speech.org/archive/interspeech%5C_2010/i10%5C_2014.html (cit. on p. 11).

[37] L. F. Gallardo and B. Weiss, 'Speech likability and personality-based social relations: A round-robin analysis over communication channels', in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, N. Morgan, Ed., ISCA, 2016, pp. 903–907. DOI: 10.21437/Interspeech.2016-459. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-459 (cit. on p. 11).

[38] S. Gonzalez and X. Anguera, 'Perceptually inspired features for speaker likability classification', in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, IEEE, 2013, pp. 8490–8494. DOI: 10.1109/ICASSP.2013.6639322. [Online]. Available: https://doi.org/10.1109/ICASSP.2013.6639322 (cit. on p. 11).

[39] M. Parker and S. Borrie, 'Judgments of intelligence and likability of young adult female speakers of american english: The influence of vocal fry and the surrounding acoustic-prosodic context', *Journal of Voice*, vol. 32, pp. 538–545, Sep. 2017. DOI: 10.1016/j.jvoice.2017.08.002 (cit. on p. 11).

[40] B. Weiss, J. Trouvain, M. Barkat-Defradas and J. J. Ohala, *Voice attractiveness: studies on sexy, likable, and charismatic speakers*. Springer, 2021 (cit. on pp. 11, 12).

[41] B. W. Schuller, S. Steidl, A. Batliner, E. Nöth, A. Vinciarelli, F. Burkhardt, R. van Son, F. Weninger, F. Eyben, T. Bocklet, G. Mohammadi and B. Weiss, 'The INTERSPEECH 2012 speaker trait challenge', in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, pp. 254–257. [Online]. Available: `http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_0254.html` (cit. on p. 11).

[42] M. Morise, F. Yokomori and K. Ozawa, 'Building a database for likability evaluation of uttered speech', *Acoustical Science and Technology*, vol. 41, no. 1, pp. 423–424, 2020. DOI: `10.1250/ast.41.423` (cit. on p. 11).

[43] S. M. Hughes, M. A. Harrison and G. G. Gallup, 'The sound of symmetry: Voice as a marker of developmental instability', *Evolution and Human Behavior*, vol. 23, no. 3, pp. 173–180, 2002, ISSN: 1090-5138. DOI: `https://doi.org/10.1016/S1090-5138(01)00099-X`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S109051380100099X` (cit. on p. 12).

[44] S. A. Collins and C. Missing, 'Vocal and visual attractiveness are related in women', *Animal behaviour*, vol. 65, no. 5, pp. 997–1004, 2003 (cit. on p. 12).

[45] K. P. Murphy, *Machine learning - a probabilistic perspective*, ser. Adaptive computation and machine learning series. MIT Press, 2012, ISBN: 0262018020 (cit. on pp. 12, 13, 18–20, 43).

[46] R. Rojas, *Neural Networks - A Systematic Introduction*. Springer, 1996. [Online]. Available: `http://page.mi.fu-berlin.de/%5C%7Erojas/neural/` (cit. on p. 13).

[47] S. Linnainmaa, 'Algoritmin kumulatiivinen pyöristysvirhe yksittäisten pyöristysvirheiden taylor-kehitelmänä [the representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors]', Master's thesis, University of Helsinki, 1970 (cit. on p. 13).

[48] D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning representations by back-propagating errors', *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: `http://dx.doi.org/10.1038/323533a0` (cit. on p. 13).

[49] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org (cit. on pp. 14, 17, 21).

[50] Y. Bengio, P. Y. Simard and P. Frasconi, 'Learning long-term dependencies with gradient descent is difficult', *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181. [Online]. Available: https://doi.org/10.1109/72.279181 (cit. on pp. 14, 15).

[51] S. Hochreiter and J. Schmidhuber, 'Long short-term memory', *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997 (cit. on pp. 14, 15).

[52] F. A. Gers, J. Schmidhuber and F. A. Cummins, 'Learning to forget: Continual prediction with LSTM', *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000. DOI: 10.1162/089976600300015015. [Online]. Available: https://doi.org/10.1162/089976600300015015 (cit. on p. 15).

[53] A. Graves and J. Schmidhuber, 'Framewise phoneme classification with bidirectional LSTM and other neural network architectures', *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005. DOI: 10.1016/j.neunet.2005.06.042. [Online]. Available: https://doi.org/10.1016/j.neunet.2005.06.042 (cit. on p. 15).

[54] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger and R. Shah, 'Signature verification using A "siamese" time delay neural network', *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993. DOI: 10.1142/S0218001493000339. [Online]. Available: https://doi.org/10.1142/S0218001493000339 (cit. on p. 16).

[55] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition', in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556 (cit. on p. 17).

[56] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, P. L. Bartlett, F. C. N. Pereira, C. J. C.

Burges, L. Bottou and K. Q. Weinberger, Eds., 2012, pp. 1106–1114. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (cit. on p. 17).

[57] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90 (cit. on p. 17).

[58] T. Mikolov, K. Chen, G. Corrado and J. Dean, 'Efficient estimation of word representations in vector space', in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781 (cit. on p. 17).

[59] M. Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*. USA: John Wiley & Sons, Inc., 2002, ISBN: 0471228524 (cit. on p. 18).

[60] R. Kohavi, 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, Morgan Kaufmann, 1995, pp. 1137–1145. [Online]. Available: http://ijcai.org/Proceedings/95-2/Papers/016.pdf (cit. on p. 20).

[61] N. Buduma and N. Locascio, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*, 1st. O'Reilly Media, Inc., 2017, ISBN: 1491925612 (cit. on pp. 21, 22).

[62] G. Heigold, I. Moreno, S. Bengio and N. Shazeer, 'End-to-end text-dependent speaker verification', in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, IEEE, 2016, pp. 5115–5119. DOI: 10.1109/ICASSP.2016.7472652. [Online]. Available: https://doi.org/10.1109/ICASSP.2016.7472652 (cit. on p. 22).

[63] C. Jemine, *Resemblyzer*, https://github.com/resemble-ai/Resemblyzer, Accessed: 2021-04-18 (cit. on p. 23).

[64] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, 'Librispeech: An ASR corpus based on public domain audio books', in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, IEEE, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964 (cit. on p. 23).

[65] A. Nagrani, J. S. Chung and A. Zisserman, 'Voxceleb: A large-scale speaker identification dataset', in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed., ISCA, 2017, pp. 2616–2620. [Online]. Available: http://www.isca-speech.org/archive/Interspeech%5C_2017/abstracts/0950.html (cit. on p. 23).

[66] J. S. Chung, A. Nagrani and A. Zisserman, 'Voxceleb2: Deep speaker recognition', in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed., ISCA, 2018, pp. 1086–1090. DOI: 10.21437/Interspeech.2018-1929. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1929 (cit. on p. 23).

[67] J. Shor and O. Lang, *Improving speech representations and personalized models using self-supervision*, https://ai.googleblog.com/2020/06/improving-speech-representations-and.html, Accessed: 2021-04-18 (cit. on p. 24).

[68] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal and M. Ritter, 'Audio set: An ontology and human-labeled dataset for audio events', in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, IEEE, 2017, pp. 776–780. DOI: 10.1109/ICASSP.2017.7952261. [Online]. Available: https://doi.org/10.1109/ICASSP.2017.7952261 (cit. on p. 24).

[69] R. Z. Jiménez, L. F. Gallardo and S. Möller, 'Scoring voice likability using pair-comparison: Laboratory vs. crowdsourcing approach', in *Ninth International Conference on Quality of Multimedia Experience, QoMEX 2017, Erfurt, Germany, May 31 - June 2, 2017*, IEEE, 2017, pp. 1–3. DOI: 10.

1109/QoMEX.2017.7965678. [Online]. Available: https://doi.org/10.1109/QoMEX.2017.7965678 (cit. on p. 27).

[70] T. Baumann, *Experiment zur Sprecherpräferenz*, https://www.timobaumann.de/temp/beaqle/, Accessed: 2021-04-18 (cit. on p. 27).

[71] R. Herbrich, T. Minka and T. Graepel, 'Trueskill$^{\text{tm}}$: A bayesian skill rating system', in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, B. Schölkopf, J. C. Platt and T. Hofmann, Eds., MIT Press, 2006, pp. 569–576. [Online]. Available: https://proceedings.neurips.cc/paper/2006/hash/f44ee263952e65b3610b8ba51229d1f9-Abstract.html (cit. on p. 29).

[72] B. Milde and C. Biemann, 'Unspeech: Unsupervised speech context embeddings', in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed., ISCA, 2018, pp. 2693–2697. DOI: 10.21437/Interspeech.2018-2194. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-2194 (cit. on p. 62).

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Studiengang Bachelor Mensch-Computer-Interaktion selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel — insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen — benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ort, Datum

Unterschrift

Erklärung zu Bibliothek

Ich bin damit einverstanden, dass meine Arbeit in den Bestand der Bibliothek eingestellt wird.

Ort, Datum	Unterschrift