Unsupervised Learning for Expressive Speech Synthesis

Doctoral Thesis

—

Author: Igor Jauk
Supervisor: Antonio Bonafonte

June 29, 2017

Alea iacta est. IVLIVS CAESAR

Abstract

Nowadays, especially with the upswing of neural networks, speech synthesis is almost totally data driven. The goal of this thesis is to provide methods for automatic and unsupervised learning from data for expressive speech synthesis. In comparison to "ordinary" synthesis systems, it is more difficult to find reliable expressive training data, despite huge availability on sources like Internet. The main difficulty consists in the highly speaker- and situation-dependent nature of expressiveness, causing many and acoustically substantial variations. The consequences are, first, it is very difficult to define labels which reliably identify expressive speech with all nuances. The typical definition of 6 basic emotions, or alike, is a simplification which will have inexcusable consequences dealing with data outside the lab. Second, even if a label set is defined, apart of the enormous manual effort, it is difficult to gain sufficient training data for the models respecting all the nuances and variations.

The goal of this thesis is to study automatic training methods for expressive speech synthesis avoiding labeling and to develop applications from these proposals. The focus lies on the acoustic and the semantic domains. For the part of the acoustic domain, the goal is to find suitable acoustic features to represent expressive speech, especially for the multi-speaker domain, as getting closer to real-life uncontrolled data. For this, the perspective will slide away from traditional, mainly prosody-based, features towards features gained with factor analysis, trying to identify the principal components of the expressiveness, namely using i-vectors. Results show that a combination of traditional and i-vector based features performs better in unsupervised clustering of expressive speech than traditional features and even better than large state-of-the-art sets in the multi-speaker domain. Once the feature set is defined, it is used for unsupervised clustering of an audiobook, where from each cluster a voice is trained. Then, the method is evaluated in an audiobook-editing application, where users can use the synthetic voices to create their own dialogues. The obtained results validate the proposal.

In this editing application users choose synthetic voices and assign them to the sentences considering the speaking characters and the expressiveness. Involving the semantic domain, this assignment can be achieved automatically, at least partly. Words and sentences are represented numerically in trainable semantic vector spaces, called embeddings, and these can be used to predict the expressiveness to some extent. This method not only permits fully automatic reading of larger text passages, considering the local context, but can also be used as a semantic search engine for training data. Both applications are evaluated in a

perceptual test showing the potential of the proposed method.

Finally, accounting for the new tendencies in the speech synthesis world, deep neural network based expressive speech synthesis is designed and tested. Emotionally motivated semantic representations of text, sentiment embeddings, trained on the positiveness and the negativeness of movie reviews, are used as an additional input to the system. The neural network now learns not only from segmental and contextual information, but also from the sentiment embeddings, affecting especially prosody. The system is evaluated in two perceptual experiments which show preferences for the inclusion of sentiment embeddings as an additional input.

Acknowledgements

First of all I would like to thank Antonio Bonafonte for his help, lead and patience, and for the opportunity to work and to develop this work in his group.

Next, I would like to mention the FPU grant (Formación de Profesorado Universitario) from the Spanish Ministry of Science and Innovation (MCINN) which made possible the research documented in this thesis as well as the short-term stay at the University of El Paso, where part of this work, related to Chapter *Semantics-to-Acoustics Mapping*, was designed and implemented. At the same time I would like to thank Prof. Nigel Ward from the University of Texas at El Paso for his friendly receive and advise in mentioned topic. Further, I would also like to mention the NII International Internship Program which made possible the short-term stay at the National Institute of Informatics (NII) in Tokyo and to thank Prof. Junichi Yamagishi for hosting me and supervising the research related to Chapter *NN-based expressive speech synthesis with sentiment embeddings*.

I would like to thank for all the additional help received from many persons on the way to the finish line, among them Dani, Carlos, Santi, Sergi, Jaime, Lauri, Xin, Shinji, Paula, Gustav, all the participants who suffered in my listening tests and everybody else who helped and encouraged me, as well as anybody who I should mention and forgot.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech synthesis is an old, almost romantic, idea of machines, computers, and robots, who talk and express themselves as do human beings. In futuristic science fiction movies and literature, there is almost no way around a talking computer or robot. Sometimes, the talking computer, despite of the vast artificial intelligence capabilities, is identified as such, talking in a robotic and monotonous voice, and being nothing else than an aid to humans, like for instance in the *Star Trek* movies the "Computer" is an important tool for the space ship crew. In different occasions, computers act very human-like, imitating emotions and free will, like for instance in *2001: A Space Odyssey* the computer tries to kill the crew, or the very same-type computer tries to seduce Marge and to kill Homer in *The Simpsons*; or in *Her* a *Siri*-like personal assistant basically substitutes a life-partner. Sometimes even, robots mean to lead humans astray passing for human beings for a variety of reasons like killing them, like in *The Matrix* or *Terminator*; living among them and being accepted as intelligent beings, like in *I, Robot* and the book series by Asimov; serving as life-partner or children substitutes, like in *Ex-machine* or *AI*; or just being an integral part of the society, like in *Star Wars*. In some of the stories, machines are just part of everyday life, while in others, they are the central aspect. In all cases it seems that the interpretation of intelligence in computers goes hand in hand with the capability to speak **like** human beings, i.e. emotionally and expressively. This aspect is ironic because often, superior intelligence in human beings is being characterized as cold and emotionless, like for instance *Mr. Spock* in Star Trek, or *Sheldon Cooper* in *Big Bang Theory* who is anxious in imitating Spock.

Taking a look at the reality, talking machines are not yet so far as to pass for humans, even aside from the aspect of the (artificial) intelligence, but it is not left unattempted. First known experiments with talking machines date back as far as to the 18th century, where mechanical vocal tract models were built to produce speech-like sounds. In 20th and 21st centuries, the computerization and digitalization opened totally new possibilities in research, and the so called *speech synthesis* was invented. When computers gained performance and power, statistical learning became of great importance, and speech synthesis, among a vast number of other research areas, has experienced an incredible boom, leading to astonishing results.

Nevertheless, although speech synthesis has already achieved very high intelligibility and naturally sounding, sometimes hardly distinguishable from natural human speech, a talking computer still would not be able to pass for a human, putting apart artificial intelligence. Why? Because sounding naturally and intelligibility alone is not enough.

At this point, one could ask: Why actually should we want a computer to sound like a human? Why not let it sound like a computer and forget all the troubles and science fiction fairy tales? Well, let's take a very natural sounding synthetic voice of a state-of-the-art synthesis system and making it read an E-mail or a short message. Probably, it will yield satisfactory results. However, taking the same voice and making it read a book or dub a film is a totally different task. Probably, after some minutes of listening the voice will sound boring and monotonous. Basically, "natural-sounding" is not enough. *Expressiveness*, which is adapted to each possible situation in an oration, is indispensable. Actually, even a human reader needs to be expressive in such a task. And this is exactly the point why we do want synthetic voices to sound expressive, to substitute humans in tasks like book reading, film dubbing, and maybe one day, when machines become intelligent, to aid humans in space odysseys and accompany them in everyday life.

Expanding the idea of expressiveness, as well as a very human-sounding but monotonous voice is inappropriate for book reading or film dubbing, an expressive voice which uses expressiveness in a wrong way, also would not satisfy the listener. If, for instance, a book character should sound sad, but sounds happy, destroys the impression and transmits a wrong message. Of course, there might be a person who applies the correct voice to each situation, but wouldn't it take us a step closer to the science fiction world if the machine could do it by itself..?

Now, how can we make a speech synthesis system sound expressive? To answer this question, we first need to know, what is actually "sounding expressively". It is not enough just to dispose of a happy or a sad voice, we also need to know, when to use it. And also, we would need many more voices beside the sad and the happy one. Of course, we also could add an angry one, and others like bored, surprised, disgusted, and so on. But, what if in a book, a giant gets angry. Will he sound the same as an old lady, even apart of the fact that he should have a different voice? Probably not. Also, a surprise can be positive or negative, it can be sad or happy, angry, cold, hysterical, and so on. Basically, all emotions, expressions, speaking styles are speaker and situation dependent, influenced by intention and attitude, are gradually variable, and can be mixed in between them. This means that a few defined emotional voices are basically not enough for a realistic book reading or film dubbing.

Given the practically unlimited number of all possible expressions, how can we account for all of them by the available means in speech synthesis? To answer this question, we need to dive on a deeper level and ask the question, how is expressiveness represented acoustically and textually? When we hear an expressive voice, how do we know that it is sad or happy? Certain acoustic characteristics must carry this information such that it is audible. On the other hand, situational pragmatics are very important in order to fully understand the underlying expressiveness. In the speech synthesis task, the input to the system is generally plain text, with no further information about how it has to be said.

How can we deduce information about expressiveness from plain text and use it effectively as input for the system in order to modify acoustic characteristics of the synthetic voice such that it sounds not only expressive, but appropriate for the input text?

A different issue in building expressive speech synthesis is the work load. Most of the work goes in data acquirement or preparation, such as recordings, labeling, and so on. It is reasonably difficult to find annotated sources of expressive speech, and even if they can be found, the number of expressions is generally very limited. On the other hand, to label a rich source of expressive speech such as audiobooks, films, dialogues, etc., is not only very costly, but labels are also difficult to define due to the almost infinite number of possible expressions.

1.1 Thesis Goals

Emanating from this reasoning we can proceed to formulate main thesis hypothesis and the overall goals of the thesis with some key ideas of how to achieve them.

1. It is possible to define expressive voices from clusters of data in the acoustic domain, applying unsupervised methods to build the clusters, i.e. no labels of human interpretation are permitted to define the voices or the data in the clusters.

2. It is possible to improve the expressiveness of a synthetic voice using in the training process semantic features which codify some sort of expressive information and are obtained fully automatically.

These two main hypothesis clearly aim at the two domains which are in focus of this investigation: The acoustic domain and the semantic domain. The deriving main research questions regarding both domains are:

1. How is expressiveness represented in the acoustic and the semantic domains?

2. Is it possible to define reliable features for each of the domain which reflect the expressiveness?

3. If relevant features can be defined, how can they be used to prove the hypothesis in each case?

In the course of the thesis, arguments and experiments will be presented which are oriented towards answering and approving or disapproving the hypothesis. For now, these relatively free hypothesis and questions will be substantiated in a series of concrete theses goals and key ideas.

1. We want an expressive speech synthesis system, i.e. a system that has a repertoire of expressions, speaking styles, and emotions which can be recognized in the output of the system.

- We need a state-of-the-art synthesis method which can be trained flexibly and be able to produce reasonably good speech quality. We can train the system such that it synthesizes speech with different expressions.

2. The number of all possible expressions, emotions or speaking styles must not be limited in order to cope with the "unlimited" number of real-life expressions, emotions or speaking styles. The consequence of this is that, basically, hard classifying and labeling must be avoided. Also, the system must be adaptable to new databases and new requirements in voice and expression repertoire.

   - We need a method to automatically define training data for expressions, based on acoustic features or semantic content of the training data.
   - The synthesis system must be flexible enough as to change its expressive state according to the current requirements or to update its expressive repertoire regarding current requirements. In this case, traditional labels would be substituted by automatically derived numerical representations.

3. The system must be usable in an appropriate way in each situation, i.e. use the right expression at the right time, in a manual, and if possible, in an automatic way.

   - First, system's expressive repertoire must contain the adequate number of expressions or be able to acquire them, so they can be chosen manually. Second, methods must be developed to derive expressiveness from text such that the system automatically could determine which expression to use, or, if necessary, to automatically choose training data to acquire a new expression.

4. The work load to acquire data and train the system must be minimal, and as many processes as possible must be automatized.

   - Training data must be selected automatically. For this, unsupervised methods must be available to cluster data, based on acoustic or on semantic features. Alternatively, the system must intrinsically learn from data which acoustic characteristics must have the output regarding some expressiveness-related input characteristics.

Intuitively, the most appropriate technical methodology for these goals must be related to statistical modeling in order to learn the required properties from data, allowing a significant reduction of the work load. In state-of-the-art systems, statistical learning is a key feature for at least some of the tasks a synthesis system must perform. As will be shown on the course of the present work, these methods can be successfully applied to achieve the formulated goals.

1.2 Thesis Overview

The thesis will have the following structure, regarding all the different aspects named above. First, in order to understand the key aspects of expressive speech

synthesis, it is necessary to study and comprehend how speech synthesis actually works, and how its composing elements can be used for expressive speech. Chapter 2 gives an overview of the integral elements of a speech synthesis system and how can they be used for expressive speech. It discusses the main synthesis techniques, from the traditional ones, to the newest state-of-the-art methods, including an introduction to *deep learning*. Finally, it discusses the term "expressive speech" and gives a state-of-the-art overview of expressive speech synthesis developed so far.

As stated, numerical representation is of great importance in order to analyze data and create training corpora. Chapter 3 treats the problem of the acoustic feature representation of expressive speech. Traditionally, the focus lies on prosodic features in the expressive field. However, various studies have shown that prosodic features alone are not enough to represent all types of expressions. Also, the task gets more difficult in multi-speaker databases like audiobooks. For these reasons, different feature sets have been proposed, including prosodic and spectral features, musical notes, voice quality features, and more. In speaker recognition, a new type of discriminative feature has been proposed, the i-vectors. I-vectors represent speakers in form of vectors, where the vectors of the same speaker are close to each other in the vector space, while those of different speakers are far away. This technique is proposed here also for expressive speech analysis in multi-speaker databases. Chapter 3 compares i-vector based feature sets to traditional features, and other state-of-the-art sets used in emotional speech analysis, showing that i-vectors outperform all the other feature sets in the multi-speaker domain.

Continuing the examination of expressive speech, Chapter 4 discusses the possible textual-based, semantic representation of expressiveness. Here, the focus lies on vector representation of text semantics. Similar to i-vectors, semantic vectors represent textual units, e.g. words or sentences, in a vector space, where semantically similar units are close to each other, while semantically distant units are far away from each other in the vector space. Different methods of representation are discussed, starting from *singular value decomposition* and ending up with representations generated by neural networks. These latter ones can be adjusted to specific needs manipulating the training criterion, opening many ways to gain efficient representations. The vector representations obtained from text are used to predict acoustic representation of expressiveness, allowing for automatic data clustering of expressive speech based on semantics of the underlying text. This ability is explored to create emotional voices in a semi-supervised manner, or to read book paragraphs with expressive voices in a fully automatic way, where for each sentence an individual synthetic voice is trained. Finally, experiments and results on the subject are presented.

Clustering of expressive speech is a powerful method to create training data, however, no matter if it happens on acoustic or on semantic level, it has to rely on the underlying extracted features. Since feature extraction will always contain an error, there will be always lost and added data in clusters. Using neural network based synthesis, data clustering is unnecessary since the networks adjust their parameters on the basis of the complete dataset, regarding any given input feature. Chapter 5 presents a study where semantic vector representations of the input text, trained on sentiment of sentences (i.e. positiveness or negativeness), are used as an additional input to a neural network based speech synthesis

system. The system learns effectively from the vectors and adapts the output depending on the sentiment. Experiments show that especially prosodic values, particularly pitch, are strongly affected by the sentiment vectors.

Finally, a conclusion will be drawn regarding studies carried out in the present thesis, the difficulties and the challenges of the task, as well as possible future work regarding the panorama of newly emerged synthesis techniques based on neural networks.

Chapter 2

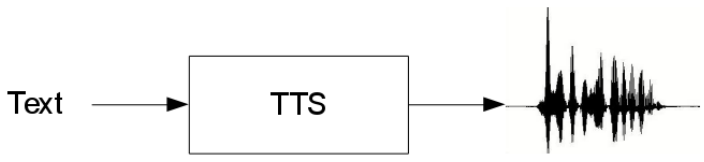
Speech synthesis: Review and State-of-the-Art

Speech synthesis is a computational technique of producing synthetic, human-like speech by computer. Normally, the input to a speech synthesis system is plain or marked-up text, hence the name *text-to-speech synthesis (TTS)* (see for instance Sotscheck et al. (1996)), which is then analyzed and converted to speech. The analysis process involves the deduction of a series of characteristics which are used to derive the waveform, as illustrated in figure 2.1.

TTS systems are very common in use, for example, for visually impaired persons, speech impaired persons, in automatic translation applications (for instance *Google* translator), E-mail reading, personal assistants, etc.

In a TTS system, the user has direct control of what is being said, but rarely of *how* it is to be said. The *how* something is said involves prosodic aspects like speech melody, rate, etc., but also more pragmatic high-level facets like expressiveness, speaking styles and emotions.

Figure 2.1: Text-to-speech system.



There are several underlying conceptual and technical problems in TTS systems which need to be discussed and taken into account. In order to understand them it is necessary to understand how human speech production works and what it is good for. Speech is mean of communication such that some sort

of information is transmitted between speakers. As for instance Taylor (2009) argues, there are three basic types of communication: *affective*, *iconic*, and *symbolic* communication.

The affective communication is the most basic type, where emotions and instincts are communicated, like for instance *yelling* as expression of pain or *laughing* as expression of joy.

Iconic communication is a communication form where forms are defined, which resemble the meaning of the intended communicational goal. For example, a road sign where slippery roads are indicated by a drawing of a skidding car. Also visual art like painting could be seen as an iconic form of communication.

Symbolic communication is the most complex one since it works with abstract symbols, such as letters or phonemes, which form is generally not related to meaning. These symbols can be combined to larger units, like words, sentences, etc., so an arbitrary number of meanings can be transmitted. Human languages, in spoken and in written form, are generally symbolic communication systems. Some writing systems have acquired a totally abstract form, such as semi-phonetic alphabets as the Roman or the Cyrillic alphabets. Other writing systems clearly conserve an iconic origin, such as Chinese.

Language, as a communication system, is an organized and intended way of transmitting information. Historically, the primary form is the spoken form. Many neurological and physiological processes are involved in order to create a speech signal (see for instance Geschwind (1972); Schade (1999)). First, an idea or a concept originates in the brain, involving processes like memory activation, conceptualization, then, nerves for muscle control are activated, etc., all this being a highly complex process by itself. On the physiological side, in order to produce a signal, the air flow, which originates in lungs, passes through the *glottis*, where, for voiced parts of the signal, the glottal folds close and are brought to vibration by the passing air flow, as postulated by the *myoelastic* and the *aerodynamic* theories, as explained for instance by Van den Berg (1958). Then, the resulting signal is filtered by the vocal tract according to the *acoustic theory of speech production*, as in Fant (1970); Ungeheuer (1962). Different articulatory positions of speech organs cause different physical conditions for air flow, changing resonance frequencies and allowing for forming of phonemes. The physical imperfection of the vocal tract causes lateral effects, such as energy loss, perturbations, and so on, modifying the signal, and actually making it sound "natural".

In comparison to the biological process, speech synthesis is intrinsically different. First, in a TTS system, text is primary since it constitutes the input to the system. Second, since machines have no articulation organs, there is need for a mathematical model to generate voice-like sounds, also taking into account the mimicking of those "imperfections" of the signal which make human speech sound "natural". Finally, there is need for a model which connects both parts, the text and the sound. This is tricky because of the totally distinct nature of both signals. Text is visual (except for *braille*, which is tactile), discrete and permanent, while speech is a continuous, non-permanent, acoustic wave.

In essence, TTS is used for symbolic communication. However, *expressive* TTS is actually a crossing of symbolic and of affective communication. Information needs to be transmitted as in symbolic communication, but it needs to be shaped

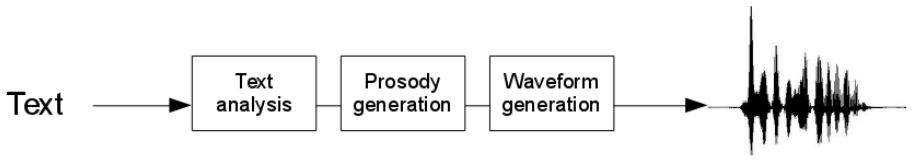
by an underlying affective form, which is interpretable by humans. This affective form is natural in inter-human communication, and between animals in general, however, why a computer needs to convey *happiness*, or *sadness*, or *confusion*, or whatever other type of affective information? The answer to this question is the same as in the general case of speech synthesis: to substitute humans in certain applications. For instance, book reading for children needs to be expressive, if not, they get bored and won't listen. Or dubbing, or translation, basically, all type of speech that goes beyond the simple reading of an E-mail or message.

In order to understand how we can teach expressiveness to a machine, first, it is necessary to understand, how can we generally teach a machine to speak. This chapter focuses on the introduction of how speech synthesis works, and how these common techniques can be used to introduce the "affective" component to the system. Section 2.1 briefly presents the general aspects of a text-to-speech synthesis system. It focuses on text analysis, prosody prediction, and early waveform generation methods. Text analysis extracts from text information, useful to deduce acoustic properties, which are also important for expressiveness. Prosody, as will be discussed later, is considered traditionally as the main correlate of expressiveness in the acoustic domain, such that it is of most importance to understand how prosody prediction works and how it can be applied to expressive speech synthesis. Early waveform generation methods help to understand the basic ideas of how TTS systems work, and it will be shown that even the most primitive (computer-based) mechanisms were used to create expressive speech. Section 2.2 introduces unit selection TTS systems. These systems are still considered to be a reference and are still widely used in expressive speech synthesis. Section 2.3 describes statistical, HMM-based systems. These systems provide great flexibility and a robust speech quality. A large part of the present work uses HMM-based systems to synthesize expressive speech. Section 2.3.1 introduces speaker adaptation techniques which allow for voice training with little amounts of data in statistical synthesis. These techniques are very important in expressive speech synthesis because of the sparsity-of-data problem. The number of all possible expressive styles is practically unlimited, which makes it impossible to gain enough training data for all the expressions. Speaker adaptation techniques allow to compensate this problem. Section 2.4 introduces the deep learning paradigm and in section 2.4.2 state-of-the-art neural network based systems will be presented and discussed. Neural network (NN) based speech synthesis is relatively novel and in the last years has experienced a "boom" in the TTS community, outperforming other techniques. However, due to the novelty of the approach, little work has been invested in expressive speech synthesis with neural networks. This work will introduce some of the first experiments on expressive speech with neural network based speech synthesis systems, as will be presented in Chapter 5. Finally, section 2.5 presents a review of expressive TTS systems and the techniques which have been used to create them.

2.1 General notions of TTS systems

Figure 2.2 shows the three basic modules of TTS system. What happens in a TTS system? First, written text is introduced as system input.

Figure 2.2: Text-to-speech system divided in three modules.



Written text generally encodes information about *what* is said, with very limited information about *how* it is said. *What is said* refers to the segmental information which constitutes semantic encoding. The main difficulty about it is the arbitrariness and ambiguity of textual units, e.g. "a" is pronounced as /ɑː/ in *car*, as /æ/ in *drag*, and not pronounced at all in *read*. Further, there might be differences to regional realizations of the same unit, e.g. *dance* as /d æn s/ in American English and /dɑːns/ in British English, according to the Cambridge dictionary. Additionally, there are languages, where no pronunciation can be deduced from writing, like Chinese or Japanese. For instance in Japanese each *Kanji* can have dozens of different ways of pronunciation with no clear rules which one to choose in which situation, like for instance, according to *Rōmaji Desu* among others, 行 (to go) can be read as /i.ku/[1], /ju.ku/, /juki/, /ju.ki/, /iki/, /i.ki/, /okona.u/, /oko.nau/, /koː/, /gʲoː/, /an/, and also vary in exceptional cases, like for example the pronunciation of 行る (to send) is /ja.ru/.

How it is said refers to prosodic and extra-/paralinguistic information such as sentence accent, focus, speaking style, and so on. There are little cues in text about this type of knowledge. Punctuation marks give hints about sentence modus and breaks. Everything else must be deduced from text semantics, context, or world knowledge. In practice, most expressive speech synthesis systems make it user's choice, which expressive style to synthesize. Sometimes the user can just choose the emotion or the expressive voice, also the input text can be marked up such that additional information can automatically be interpreted by the system. Also, as will be seen further, automatic methods exist which can be used to deduce expressiveness from plain text.

The final step is the actual waveform generation. Different techniques have been applied to this problem, including knowledge-based signal processing techniques, corpus-driven concatenation approaches, statistical and neural network based approaches. All of them have been used for expressive speech synthesis, with their particular advantages and disadvantages.

No matter which synthesis method is applied, there is always need for data. In

[1]The point "." in the IPA transcription means that the actual Kanji is pronounced according to the left side of the transcription separated by the point, the right side must be added as an additional syllable, being the pronunciation of the Kanji part of this concrete combination, and not transferable to combinations with other syllables.

the case of formant synthesis or similar techniques, it can be used to deduce rules; in the case of concatenation techniques, it is the source of concatenation units; in the case of machine learning techniques, it is the source of information for the models. Corpus preparation and feature extraction is an important first step and often the most laborious one.

The next section introduces text analysis techniques which are used to gain information from plain text. Afterwards, prosody prediction techniques are addressed, whereas generally, since prosody needs to be predicted from text, information gained in text analysis is used for prosody prediction. Then, corpus preparation will be discussed shortly. Corpus preparation often combines text analysis and feature extraction techniques, applied a priori and off line on large amounts of data. On continuation, waveform generation techniques will be discussed, with mentions of corresponding expressive speech synthesis systems. A special focus will be put on concatenation and unit selection methods in Section 2.2 since these systems still comprise state-of-the-art and reference quality for current TTS. Then, in Section 2.3, statistical techniques will be introduced and discussed. In Section 2.4, deep learning synthesis methods will be addressed. Finally, Section 2.5 will discuss expressive speech synthesis systems and their methods.

2.1.1 Text Analysis

Everything related to gaining information from text is referred to as *text analysis* or *text (pre-)processing*. Text analysis is a crucial first step in all TTS systems since it is almost always the only input which a TTS system has. So basically, all information, segmental, prosodic, expressive, must be gained from text analysis, unless in some other way provided by the user. The goal is to create a representation of the input text such that the prosody can be predicted, the expressiveness can be deduced, and segmental models get the necessary information to generate speech. The following is an (incomplete) list of possible text analysis steps.

- **Text format**: Plain text, tables, lists, XML, dialogue, etc. It is essential to regard the text format of the input text and convert it to the internal text format, if necessary. The text can be enriched with significant information about the pronunciation, like marking the focus, speaking style, emotion etc. A well known markup language for speech synthesis is the *Speech Synthesis Markup Language (SSML)* proposed by W3C (b). It allows for encoding of possible information about the text, like prosodic information (F0, duration, etc.), structural information (breaks, focus, etc.), and also meta information (speaker, gender, etc.), in an XML format. Also, there is a specially developed markup language for emotional speech, the *emotion markup language (EmotionML)*, also proposed by W3C (a). It allows for additional marking of emotion related information, like type of emotion, attitude, affect state, action tendency, emotional vocabulary, etc., also more low-level marks like durations, timings, and other value attributes.
- **Word normalization**: Apart of the intrinsic difficulty of determining the pronunciation of textual units, as described above, there is a number

of cases where the text needs to be normalized. For example, 2/3 could be pronounced as *two third*, but also as *second of march*. The most common cases where normalization is needed are:

- *Numbers*: Dates, phone numbers, currency, time, ordinary numbers, decimal numbers, fractions, Roman numbers, etc.
- *Abbreviations and Acronyms*: Mr., UN, tel., USA, GB, NATO, etc. Some need to be expanded, some are read as words, some need to be spelled, etc.
- *Reading differences*: For instance *read* $/ri : d/$ versus $/rE : d/$.
- *Irregularities*: Especially in real-life emotional speech like dialogues, many irregular pronunciations occur, very long sounds, fillers, breaks, irregular words, etc. These segments are difficult to treat in synthesis, often they are excluded from building a system or are corrected with an orthography corrector.
- *Foreign words and proper names*: Generally need a dictionary entry with the correct pronunciation.

Word normalization usually depends on dictionaries and rule based approaches such as regular expressions, where words are analyzed and expanded, if needed. There are also automatic approaches, like for instance by Sproat et al. (2001), where non-standard words are treated systematically on a language model basis, learning the pronunciation of tokens.

- **Grapheme-to-phoneme conversion (G2P)**: Since, even in languages with phonetically oriented alphabets, the graphemes (≈ letters) do not completely correspond to phonemes, which intend to describe phonologically meaningful sounds avoiding ambiguity, a grapheme-to-phoneme conversion must be performed. The standard phoneme set was proposed by the *International Phonetic Association (IPA)* and is called *International Phonetic Alphabet (IPA)* . The alphabet provides a number of standardized symbols to describe consonants by type, voiceness and place of articulation, and vowels by grade of mouth aperture, lip roundness and position in the mouth (front to back). It also provides a set of symbols to describe special sounds like clicks and implosives and has a limited set of symbols to describe some prosodic phenomena, such as accents. Additionally, it offers diacritics to describe phonetic modification of sounds, like *voiced* or *unvoiced* pronunciation, length, tone, and so on. IPA is widely used for human-made transcriptions, however, for automatic computational processing it was impractical because of the codification. For this purpose the 7 bit ASCII based *Speech Assessment Methods Phonetic Alphabet (SAMPA)* was developed, as in Wells (1997). SAMPA is a computer-readable IPA counterpart, and in practice it is often changes and adapted to particular system and language needs.
  Automatic phonetic transcription usually involves a transcription dictionary and an automatic method for unknown words, for instance *finite state transducer (FST)* or *classification and regression tree (CART)* based, like for instance in Breuer and Hess (2010). Polyàkova (2015) explores in her work the adaptability of grapheme-to-phoneme conversion with a focus on multilingual domains and the pronunciation of foreign words.

- **Part-of-speech (POS) tagging**: Part of speech is the class a word belongs to, such as *verb*, *substantive*, and so on. POS is useful in pronunciation disambiguation, prosody prediction, and others. Nowadays, POS tagger are generally based on statistical prediction methods.
- **Structural analysis**: Often, structural and/or syntactic analysis is performed on text. Structural analysis involves the position of phonemes, syllables and words with respect to sentence begin and end, accented syllables, and so on. A famous example of relatively exhaustive structural information is the HTS (for more details on HTS see Section 2.3). Syntactic analysis identifies groups of words which belong together syntactically, like *nominal* and *verbal phrases*, and others. Such analysis often yields tree like structures where sentence parts are organized in a hierarchical manner, as in figure 2.3. Structural and syntactical analysis are usually involved in prosody prediction.
- **Semantic vector representation**: Although semantic vector representation of text is not an actual analysis form, but it is a powerful method of codifying text which can be used to identify speaking styles or expressiveness. Many speaking and expressive styles have semantic correlates. This is exploited in semantic vector space models since semantically similar content tends to appear agglomerated in the semantic vector space. From speech perspective, this property can be used to identify speaking and expressive styles. Semantic vector representation will be addressed in detail in chapter 4.

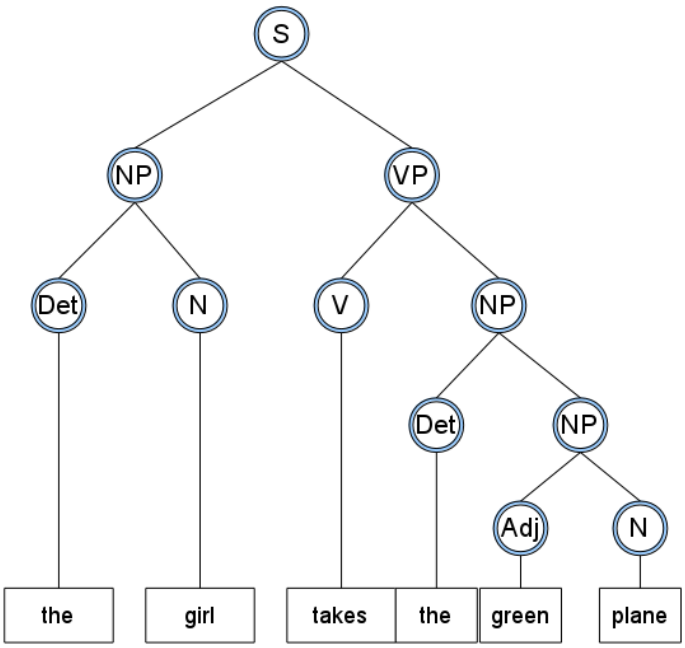
2.1.2 Prosody Prediction

Once all necessary information from text is extracted it can be used to predict the prosody. *Prosody* is a term which generally describes three suprasegmental aspects of speech: *intonation*, *rhythm* and *intensity*. Intonation is the "melody" of speech. It is one of the most functional features in speech since it is directly responsible or indirectly involved into almost all phenomena on suprasegmental level, and sometimes also on segmental level, like for instance accents and tones. Rhythm includes duration and frequency of syllables (or other segments), as well as duration and position of brakes, and structural aspects on a higher level, such as phrase accents, focus, and so on. Intensity is the "loudness" of speech. It is a suprasegmental feature, but has also a certain meaning on segmental level, for instance for accents.

As stated, prosodic features are considered to be suprasegmental features, i.e. do not directly depend on phone identity or semantic meaning of words. Rather, aspects like mode, phrase accents, stress, focus, expressiveness, speaking style, emotions, speaker identity, and so on, affect prosodic features. Good and natural prosody is crucial for naturally sounding synthetic speech and can compensate for flaws on segmental level. More details on prosodic features can be found in Chapter 3 in Section 3.1.2.

In most TTS systems prosodic components are predicted "apart" from the segmental components, which can be problematic because these two highly interconnected elements are treated separately. In expressive speech synthesis,

Figure 2.3: Binary syntactic tree structure. S: sentence, NP: nominal phrase, VP: verbal phrase, N: noun, V: verb, Det: determinant, Adj: adjective.



prosody is very important. In fact, as will be discussed later, it is often regarded as the most or even the only important feature. Regardless which method is used for expressive speech synthesis, prosody needs to be modeled carefully and respecting the expressiveness.

Intonation Prediction

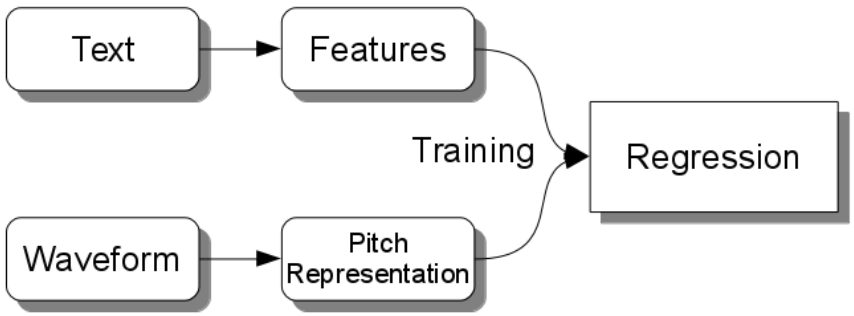
Intonation, other related terms: *melody, pitch, F0, fundamental frequency*: Intonation, acoustically, is the *fundamental frequency (F0)* of the signal. Perceptually, it is the speech melody. Articulatory, it is the effect of the vibration of the vocal folds. Well known variations of the intonation contour are: The overall declination, often attributed to the sub-glottal pressure decline, as for instance according to Taylor (2009); stress marks as pitch movements; descent at the end of a phrase except if the phrase is an echo question with a rising contour, or intermediate phrase boundary, also with risings, but weaker.

Many theories in phonology, acoustics, speech perception and production try to describe or model intonation. To name a few of them, *the Dutch Model*, by Hart and Cohen (1973), intents to stylize the contour replacing it with a series of straight lines, such that the original contour is matched as close as possible. Also contour declining is taken into account, enclosing the contour between globally declining straight lines; *Kiel Intonation Model (KIM)*, by Kohler (1991), see also Möbius (1993); Wollermann (2012), describes prosody on dif-

ferent levels, like *lexical stress*, *sentence stress*, *prosodic boundaries* in terms of pause duration, phrase-final segmental lengthening, F0 end points, *speech rate*, *intonation* in terms of "peaks" and "valleys" and their synchronization with syllables, pitch category, downstep (a concept opposed to the global declining, where pitch lowers in steps between phrase accents); *Fujisaki model*, by Fujisaki and Kawai (1982), is a superpositional model, which input is constituted by phrase and accent components which are mixed using second order filters; *Tilt model*, by Taylor (1992), defines two types of events, *accent* and *boundary*, and calculates three Tilt parameters: amplitude, duration, and a Tilt parameter which defines the general shape of the event; *Bezier polynomial coefficients*, by Escudero et al. (2002), see also Agüero and Bonafonte (2004a), define five attraction points around the contour within an accent group, i.e. the period between two accented syllables, where the first end the last ones define the beginning and the end, and the rest acts like magnets attracting the curve towards them; finally, *Tones and Break Indices (ToBI)*, by Pierrehumbert (1980), is one of the most widely used intonation models derived from the autosegmental-metrical model of intonation, as in Liberman (1975). ToBI describes the intonation contour as a combination of high and low tones, denominated as H and L. With additional symbols marking rising and falling, pitch accents, boundary tones, etc. For instance, L+H* marks a rising peak accent, H+!H* is a step down onto an accented syllable from high pitch itself, L- and H- are intermediate phrase boundary tones, and H% and L% are final boundary tones.

The models described above provide formal representation of pitch. The missing step is to predict pitch contour for a given text. This is done by learning these representations from information derived from text, as in figure 2.4.

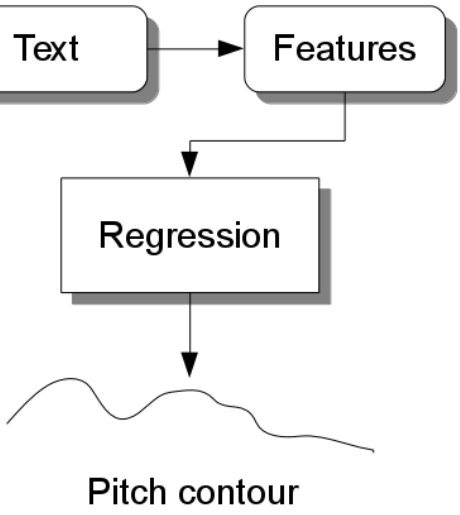
Figure 2.4: Training of a regression model for pitch prediction.



Once the model is trained, it predicts pitch contour (or a different representation) from the same type of textual features, as in figure 2.5. Typical regression models used for pitch prediction are for instance CART or DNNs. It is important that the features which are derived from text are meaningful for pitch prediction, for instance syllable/word position in phrase, distance to phrase boundary, sentence modus, accent, focus, POS, etc. In HMM- and the NN-based systems, intonation is modeled more intrinsically, using the same models for spectral and

intonation prediction. More details on intonation modeling in HMM- and in NN-based systems can be found in Sections 2.3 and 2.4.2, respectively.

Figure 2.5: Pitch prediction with a regression model.



In expressive speech, intonation can be problematic since it has higher variation and has a tendency to outliers, for instance high F0 values in angry or happy speech, or voicelessness in suspended speech. This can yield problems with the waveform generation methods, and also lead to unnatural effects in synthesized speech. On the other hand, many expressive styles use intonation in a very exhaustive way, such that it becomes an unbearable feature for expressive speech. For instance, in Chapter 3, experiments are presented where the predictive power of pitch, in comparison to other acoustic features, is very clear, at least under certain conditions. Therefore, intonation needs to be modeled very carefully, taking into account expressiveness, not only general aspects like phrase breaks.

Duration Prediction

Segmental duration is a variable factor, which on one hand depends on the phoneme type and phoneme identity, and on the other hand on the position and the function of the phoneme in the syllable, word, sentence, etc. For example, stressed vowels tend to have a higher duration then the unstressed. The vowel of the last syllable in a phrase can be several times longer in comparison to the others, which is called *final lengthening*. On the other hand, vowels in function words tend to be shorter and more assimilated, i.e. articulatory and phonetically reduced. In general, vowels are more susceptible to significant duration variations than consonants, and fricatives and liquids tend to be more variable than plosives. Finally, idiomatic factors influence the duration variabil-

ity, like for instance if duration is a distinctive feature, then it is less variable. The variability of phoneme duration is also called *elasticity*, and the *elasticity hypothesis* states that phone duration in a syllable varies due to a constant factor and is normalized by the phone class, as for instance in Taylor (2009).

In speech synthesis, duration is crucial. Segment duration needs to be predicted exactly to achieve natural sounding speech. In expressive speech synthesis, duration is as important as intonation. Depending on speaking style, duration can vary significantly, for instance shorter durations in angry or happy speech, longer durations in sad or suspended speech. Also here, outliers must be considered especially in highly aroused expressive styles like anger. In shouting or exclamations duration can vary extremely. In expressive TTS the duration must be modeled carefully regarding the expressive style and taking into account outliers.

The features which can be used to predict duration of phonemes are for instance the phoneme identity or type, presence of stress, syllable position in sentence and with respect to brakes, the type of preceding and following phonemes, the type and the position of the consonant in the syllable, and more. Some historical reference models for duration prediction are the *Klatt model*, by Klatt (1973), which is a deterministic model derived from corpus measuring where duration is calculated applying a set of factors determined by aspects like phone position, context, etc.; *Sums-of-products model*, by van Santen (1994), which is a sort of trainable Klatt model where duration is modeled as a regression; in a study by Febrer et al. (1998), the sums-of-products model outperforms the Klatt model; finally, the *Campbell model*, by Campbell (1992), which models duration on syllable level using neural networks trained on a set of linguistic features, and the phone durations are calculated according to the elasticity hypothesis, changing the durations proportionally in order to match the predicted syllable duration.

Also for duration prediction, regression techniques have been used in a similar way as for pitch prediction. Instead of pitch representations, the regression model learns segment durations from features derived from text. Duration is normally predicted on phone level and expanded to syllable and word levels, unlike the Campbell model. Also here, common regression models are decision trees and neural networks. For example, the decision tree based approach is implemented by Riley (1992) or Breuer (2009). Neural networks have been used for instance by Córdoba et al. (1999); Fackrell et al. (1999) (aside the Campbell model). Although duration is part of prosody, which is considered suprasegmental, it is mostly modeled on segmental level; rarely more global functions like rhythm are taken into account. For instance, Wagner (2008) talks about the importance of rhythm in speech; Jauk (2010) implements a network of oscillators which interpret speech rhythm as a set of pulses, where each pulse represents a rhythmic unit (syllable). In expressive speech, specific speech styles can be marked by rhythm and segment durations, like clerical speech, different types of news, etc.

In HMM-based synthesis, duration is modeled as Gaussian distribution for each state of the HMMs. In NN-based synthesis, duration can be modeled with the same models, or with a network apart. More details on duration modeling in HMM- and in NN-based systems can be found in Sections 2.3 and 2.4.2, respectively.

Break Prediction

Breaks and *pauses* are very important in speech production since they structure the general information flow, according to Wagner (2008). Partly, breaks occur due to respiratory reasons, as humans need to breath in when they lack air in the lungs. But also, breaks and timing provide a structure, which can be crucial to understanding. Therefore, some break positions are well perceived, but some others do not sound natural. Stuttering is a well known pathology which yields irregular and arrhythmic structuring of speech and affects the understanding considerably.

To give an example of appropriate and inappropriate breaking, a sentence like *"The big red elephant did not like mouse toys because they scared him."* could be uttered in different ways.

- *The big red elephant did not like mouse toys
 because they scared him.*
- *The big red elephant
 did not like mouse toys
 because they scared him.*

Other break combinations could significantly affect the understanding of the sentence though:

- *The
 big red elephant did
 not
 like mouse toys because they scared him.*
- *The big red elephant did
 not like mouse
 toys because they scared him.*

Break duration is an important factor, since too long breaks could disturb the information flow. Many breaks are not silent, but filled with noises, quasi-words, or just are formed by an intonation reset. Break position and duration depends on the word position in the sentence, on the length of the sentence, part-of-speech, distance from preceding pauses, the general syntactic structure, or physiological conditions of the speaker.

Generally spoken, breaks duration and frequency, syllable duration and word rate, are part of the rhythmic structure of speech. According to Wagner (2008); Jauk (2010), rhythm is necessary for information perception and memorizing. Also, the duration and position of rhythmical units, i.e. syllables, words, phrases, breaks, and so on, are directly related to human perception of time. For expressive speech, breaks, as intonation and duration, are crucial for the naturalness of the synthesis. Also according to Wagner (2008), there are some speaking styles which directly depend on speech rhythm, duration and breaks, for instance clerical, political or news speech.

Punctuation marks give hints for break positions and duration, however, realistic break prediction is more complex. Several approaches have been used for break prediction, including rule-based approaches, regression based approaches, finite state transducers, and so on. Agüero and Bonafonte (2004b) use a finite state transducer for break prediction. Pascual and Bonafonte (2016b) implemented an RNN to predict breaks. Agüero and Bonafonte (2003) give a review on

different break prediction methods. In natural speech, breaks are not always silences. Often, syllables are enlarged or pitch is reseted. Sometimes fillers are used, e.g. “hmmm”, “aaah”, “eeeh”; in other cases noises occur, like laughter, sighs or breathing. Adell et al. (2012) exploit the possibility to use filled pauses to augment the naturalness of synthetic speech.

2.1.3 Corpus preparation

Corpus preparation for speech synthesis is a very time-consuming and often finical issue, especially if the corpus is real-world audio data and not studio-recorded material. Normally, a raw corpus is audio data and the corresponding text. A series of procedures must be undertaken in order to use the corpus for a TTS system, either to train models, or as a unit database.

Often, as a first step, a review is necessary. For the audio part, it is very important to review the corpus for acoustically unclean data, like noise, outliers (for instance very loud or exaggerated speech), acoustic artifacts and so on. Strongly assimilated speech might be subject to removing. This is especially important in dealing with expressive speech since here, even in laboratory condition, outliers can easily occur. If unclean data is left in the corpus, it will appear in one or another form in the synthesized speech. Audio data needs to be cut, in smaller samples, normally sentence length. Also, audio data needs to have the appropriate encoding, sampling rate, etc.

For the text part too, review for unclean data needs to be undertaken. Here, unclean data can be comments, formatting marks (like HTML tags or other mark-up), misspellings and so on. It must be reviewed, if the text corresponds to what is being said in the audio. Text normalization needs to be carried out, such that numbers, acronyms, and so on are properly spelled. Finally, also here, text encoding needs to be adapted to the system requirements. All these steps are crucial when dealing with real-world data. At the end, all text, and all further information which is added, needs to be in the appropriate system format.

On continuation, a grapheme-to-phoneme conversion takes place. In the best case, a corpus specific lexicon must be created, where specific words like proper names, foreign terms or exceptions must be included in order to assure a correct pronunciation. Also, structural and syntactic analysis takes place, like number of syllables/words in a sentence, relative position of phones/syllables/words with respect to accents, phrase boundaries, etc., part-of-speech tagging, sentence parsing and other type of text analysis.

For the acoustic part, forced segmentation and labeling is performed, where speech is segmented and each segment is labeled with the corresponding phoneme (or other unit). In the past, this process was manual and very time consuming, hence only relative small corpora were used in synthesis. Nowadays, the process is generally automatic with occasional manual revision. In order to perform automatic segmentation, a speech recognition system is used, however, in comparison to the speech recognition task, in forced segmentation it is known **what** is being said. The task is to find out when exactly each unit is produced, with some allowed variations like silences after words or alternative pronunciations. This process often involves several steps, where different levels of precision

are achieved. Also, acoustic and prosodic low-level features are extracted, like MFCCs, Pitch, segmental and pause durations, and so on.

Sometimes, further labeling is required, depending on the database. For instance, in multi-speaker databases, speaker labels can identify the speakers; for emotional speech, emotion labels can be added, and so on. Generally, everything concerning manual labeling is very time consuming, and one of the goals of this thesis is to reduce manual effort in corpus preparation. More information about general aspects of corpus preparation can be found for instance in Breuer (2009). Especially challenging is the usage of audiobooks as corpora. Audiobooks are rich in expressive speech, but are recorded with different conditions than corpora intended for application in TTS. They involve character actings, imitations, exaggerations, assimilations, outliers, and so on. Some studies, for instance conducted by Chalamandaris et al. (2006) or Szekely et al. (2012), explore the problems and the solutions in dealing with audiobooks as corpora in expressive speech synthesis.

2.1.4 Waveform Generation

Waveform generation is the step where, using the information gained in previous steps, the actual speech signal waveform is generated. There are many techniques of waveform generation and generally, the kind of technique applied determines the architecture of the whole system. This section provides a brief overview of first waveform generation methods and speech synthesis systems applying these methods, partly based on a historical review offered by Klatt (1987). Sample sounds for many of the synthesizers can be found in Klatt's 'History of Speech Synthesis' Archive. State-of-the art methods are described in sections 2.2, 2.3 and 2.4.

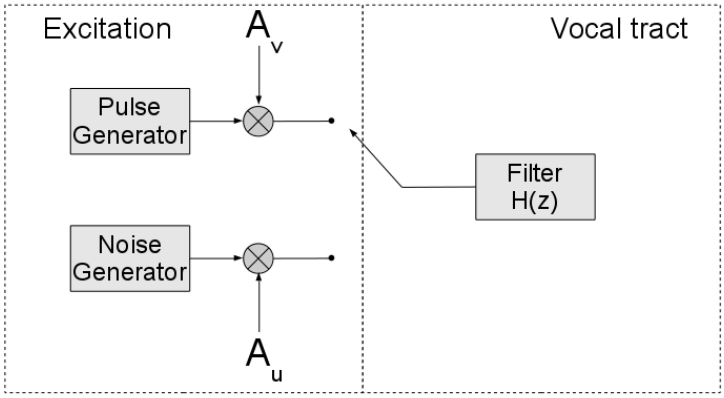
The reason why a historical insight of early methods is useful at this point is that in expressive synthesis these methods are still in use. Why? Mainly because of their simplicity and controllability. Expressiveness in speech synthesis is very variable and suprasegmental, i.e. it can be applied on all segments in general, and the acoustic effects change a lot for different expressions. For example, if we wanted a train an expressive system of 6 emotions for 6 voices, we needed a lot more training material for each voice. In more basic systems, however, like formant synthesis or diphone synthesis, all these modifications would be matter of adjustment or signal manipulation, which is far more easy than to get and prepare training data. Therefore, especially in experimental setup for fundamental research, older generative methods are still in use and will be addressed in this section. On the other hand, in general, there are not so many expressive speech synthesizers from scratch, in many cases other system types were adapted to synthesize expressive speech. Therefore, a historical perspective on mechanisms applied for expressive speech is interesting.

The earliest successful speech synthesis attempt found is documented, among others, by the Macquarie University, and goes back to 1779 to Russia, where Kratzenstein constructed a mechanical model of a vocal tract capable of synthesizing isolated vowels. The first attempt of connected speech is reported to have been conducted by Kempelen in 1791 with a pneumatic synthesizer machine, which was driven with an air flow by a whistle or an attached leather

bag which functioned as a pump. As Klatt (1987) reports, the first electronic vocoder synthesizer was built by Stewart in 1922. Dudley (1939) proposed a device which analyzed speech in terms of varying acoustic parameters, which was then able to reconstruct the original waveform, called *The Vocoder*. Also, a speech synthesizer called *Voder*, based on that technique, was published. In 1951 at the Haskins Laboratories a "Pattern Playback" synthesizer was able to convert broadband spectrogram patterns back to sound, as in Cooper et al. (1951).

An important milestone in speech synthesis was the development of the acoustic theory of speech production by Fant (1970) and Ungeheuer (1962), and the *source-filter* perspective. The source-filter perspective basically postulates that the signal generated at a source (lungs/glottis) is modified by a filter (vocal tract), as illustrated in figure 2.6. The figure shows a source part with a pulse generator and a noise generator, and a filter part with a filter corresponding to the vocal tract. The pulse generator represents the voiced part of the source signal, and the noise part represents the unvoiced part. Both have their specific amplitudes. The filter part of the system is where the formants are generated. In a mixed excitation both generators, pulse and noise generator, can be connected.

Figure 2.6: Basic source-filter speech production model with a voiced and an unvoiced component.



Based on this model, parametric *formant synthesizers* were developed. The first reported formant synthesizers were the *Parametric Artificial Talker (PAT)* by Lawrence (1953) and the *Orator Verbis Electris (OVE)* by Fant (1953). Another interesting example of formant synthesis is proposed by Klatt (1972), where the synthesizer permitted mimicking of nasalization, mixed excitation, and parallel formants for synthesis of obstruents, i.e. plosives (e.g. /p/), fricatives (e.g. /f/), and affricates (e.g. /p͡f/). The excitation and formant values are determined by hand as model parameters, which then drive a vocoder to generate the actual waveform. The advantage of the technique is the relative simplicity concerning the amount of model parameters. These can be modified by hand to produce different voices, also expressive speech. Systems developed by Cahn (1989); Murray and Arnott (1993, 1995); Burckhardt and Sendelmeier (2000) use this

technique for expressive speech synthesis, exploiting the advantage of the simplicity of adjustment. The drawback of formant synthesizers is the bad quality. Synthetic voice produced by formant synthesizers sounds "buzzy" and robotic.

Also parametric, but physiologically motivated, is the *Articulatory speech synthesis*. Here, the geometrical model of the vocal tract is emulated and an emulated source signal is modified by an emulated vocal tract according to the acoustic theory of articulation. Here, the modifiable model parameters are the geometric parameters of the vocal tract, which are translated to filter parameters so a sound wave can be generated. The first models of this kind were proposed by Dunn (1950) and Stevens et al. (1953). These first models had to be adjusted by hand for each section. A newer articulatory speech synthesis system is the *VocalTractLab* developed by Birkholz (2005). This system provides a three-dimensional visual vocal tract model which can be modified in order to simulate articulation. It has been used for several experiments and studies by synthesis on speech production, also in expressive speech, as in Birkholz et al. (2015), regarding voice quality in for instance Birkholz et al. (2011), or singing voice synthesis by Kröger and Birkholz (2009). In a very recent project, exact physical and physiological models of sound generation and of the vocal tract are used to simulate, rather than synthesize, real human voice. Until now, the technique is very expensive computationally and only vowels can be synthesized. In an example of their work, Arnela et al. (2016) study the effects of simplifying the vocal tract models on the sound production.

Generally, traditional parametric speech synthesis has always suffered from low quality, due to too simplistic modeling, like in formant synthesis or the not-optimal "classic" vocoder filters. Alternative approaches were looked for. One of the most important alternatives were systems, based on concatenation of prerecorded sound samples, that promised much better results since the generated sound was actually prerecorded human voice, hence it was "perfectly" natural. Details on concatenative speech synthesis will be discussed in section 2.2.

In the late nineties, a statistical approach, concretely *Hidden Markov Model (HMM)* based, was developed, where speech synthesis parameters were not preconfigured, but rather learned from data, as in Masuko et al. (1996). These systems have marked an important milestone in (expressive) speech synthesis, opening new possibilities. Also, new vocoder types were developed which yielded much more natural synthetic quality and renewing the importance of parametric systems opposed to the concatenative ones. These systems will be discussed in more detail in Section 2.3 generally, or in Section 2.5 regarding expressive speech synthesis. Modern state-of-the-art parametric models are based on neural networks (NN) and will be discussed in Sections 2.4 and 2.4.2. These models provide a further improvement in synthesis quality and open further possibilities.

2.2 Concatenative Speech Synthesis

The idea of the concatenative speech synthesis was to connect pieces of prerecorded speech such that variable content could be synthesized. In the most simple case, "carrier" sentences were prerecorded, where certain words or word

combinations were variable and could be substituted. Such systems were popular, and often still used, in for example train station announcements, where sentences like *Next station is:* are filled with station names. This type of systems are good for very restricted domains, but impracticable in free domains. When for example names, or any other variable content, change, and there is no recording for the new word, the person who had recorded the original content is often not available anymore, so often carrier sentences and filled words have different voices. Another important problem is the fluent concatenation and prosody. If the difference between the carrier sentence and the filled words is too big, it can affect the intelligibility.

A more ambitious idea was to record all possible phonemes of a language and combine them in order to obtain words. The problem here was the *coarticulation*. Coarticulation describes the acoustic variability in the transition between phones which is due to the movement of the articulatory organs. As for instance Menzerath and de Lacerda (1933) state, the articulation of a phone needs a temporal synchronization of distinct articulatory organs, where each has a different mass and needs to cover a different distance in order to achieve the desired position in the mouth. The consequence is that the organs do not just "jump" to certain positions in order to articulate a phone, but constantly move from one position to another, starting the movements at distinct times and with different accelerations and velocities. The main acoustic consequence is that, changing positions for consonant articulation, movements affect the resonance frequencies, i.e. formants of the vowels, such that depending on the articulatory organ and the direction of movement the formants change according to the acoustic theory of articulation. Delattre (1968) conducts extensive studies on this topic.

Besides the synchronization, there is the principle of the *economy*, as for instance according to Vary et al. (1998), which basically states that the articulatory organs try to spend less energy possible in order to achieve their goals. This same principle explains assimilations and reductions of phones, stating that while the communication goal, i.e. to transmit some information, is achieved, the articulation does not need to be perfect, i.e. the phone can be reduced.

In concatenative synthesis, the coarticulation issue could be resolved via signal processing techniques, or using better concatenation units and concatenation criteria. With the idea that the transition between two units is acoustically less stable, more difficult to model, and more important to naturalness, *diphone* based systems were developed. Diphone units did not represent a single phoneme, but rather the transition between two phonemes. For a diphone system en exhaustive corpus had to be recorded where each diphone was recorded at least once. Acoustic defects were generally corrected applying signal processing techniques. To determine the amount to be corrected, diphones were compared on acoustic, durational, and pitch levels. Since formants are variable and for many units, even by hand, difficult to determine, linear prediction coefficients were used to measure unit borders. The basic idea of *Linear prediction coding (LPC)* is that the current point in the signal can be predicted from m past points, as in Vary et al. (1998) :

$$\hat{x}(k) = \sum_{i=1}^{n} a_i x(k-i) \tag{2.1}$$

where, $\hat{x}(k)$ is the estimation of the current sample, a_i is the ith model coefficient, and $x(k-i)$ is the sample ith sample before the current one. LPCs can be used to measure the continuity between two units, and LPC filters can be interpolated in order to correct discontinuities.

For the prosodic part, intonation and durations needed to be modified according to the current needs. One of the most famous techniques for pitch and duration modification is *time-domain pitch-synchronous overlap and add (TD-PSOLA)* developed by Charpentier and Stella (1986). The idea in PSOLA was, that duration can be manipulated by copying or deleting of (invariable) pitch periods, such that the phone sounds longer or shorter. Analogously, pitch is modified by adding or removing pitch periods within the same time span, such that when more periods are produced in the same time frame, pitch is higher. This technique was very simple and very effective, however, within a limit. If changes were too big, acoustic artifacts appeared due to inconsistencies between the time- and the spectral domain.

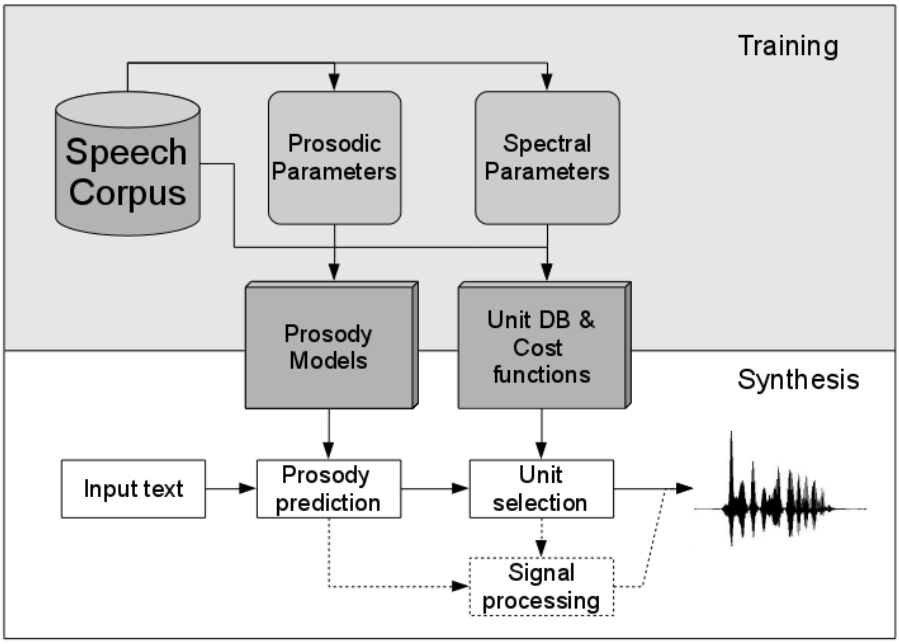
Carefully designed diphone systems had achieved reasonably good intelligibility and acceptable naturalness. Diphone systems were, and still are, also used in expressive speech synthesis, as described in section 2.5.

No matter how well designed the diphone systems were, the transitions between two phones were not enough in order to achieve real naturalness and perfect intelligibility. Applying of signal processing did not help enough, creating acoustic artifacts, though better than formant synthesis (but less flexible). Also, with cheaper memory, larger corpora could be stored and used for synthesis. Hence, a new generation of concatenative systems appeared, called *unit selection*. Unit selection systems use large databases where an algorithm tries to find the best sequence of units for each individual case, according to context, prosodic and acoustic criteria, etc. Signal processing is tried to be avoided completely. For this reason, corpora for unit selection systems must be designed very carefully and exhaustively in order to assure the availability of suitable spectral and prosodic contexts. Figure 2.7 shows a general architecture of a unit selection system. First, from a speech corpus prosodic and spectral parameters are extracted which are used to train prosodic models, and to generate the unit database and the cost functions. On the synthesis side, the input text is analyzed and the prosody is predicted. Then, a unit selection algorithm selects the best units from the unit database which satisfy the predicted and other criteria and the waveform is generated by concatenating. Additionally, signal processing methods can be applied in order to reduce small discontinuities or prosodic mismatch.

Different unit types have been used for unit selection. There are systems which use *triphones* as concatenation units. Triphones are phones in a specific phonetic context. For example [gal] is an [a] in the context of [g] and [l], [con] is a [o] in the context of [c] and [n]. Despite everything, even in very large corpora the coverage of triphones is generally not exhaustive, so similar context must be taken into account if a particular combination is missing. For instance, if the triphone [gab] is missing, it could be substituted by the triphone [gap] since [p] and [b] are articulated in the same place and have similar effect on [a].

A crucial point in unit selection systems is the concatenation algorithm. The algorithm is usually based on cost functions which decide, which units to use

Figure 2.7: Unit selection system architecture.



in each case. Two different types of costs are distinguished, *target costs* and *concatenation costs*. Target costs are costs related to the unit itself, e.g. identity, type, voiceness, phonetic context, stress, etc. Concatenation costs are related to acoustic discontinuities between two units, e.g. F0 and MFCC concatenation point discontinuities. Both types of costs must be well balanced in order to achieve good synthesis quality.

Since unfortunate concatenation yields acoustic artifacts, and especially in the early days of unit selection systems memory was expensive and corpus recording and preparation needed large effort, the idea came up that using larger units yielded better quality, reducing the number of spectral and prosodic discontinuities. However, it is impossible to dispose of an exhaustive corpus using only large units. So when large units, like words, were missing, the system could jump on lower levels and fill the gap with lower units, like syllable and phones. The problem of this type of systems is the instability and the differences in quality between passages with long and short units.

A special type of units for (non-uniform) unit selection was proposed by Breuer (2009). The units are called *phone extensions for synthesis (phoxsy)*. The main idea is that the coarticulation effects are not equal in all possible combinations, so it makes sense to define a non-uniform unit type where phones which are strongly affected by the coarticulation always remain in their context, while others, which are less affected, can be used more flexibly.

In *Ogmios* unit-selection synthesis, by Bonafonte et al. (2006), diphone-units were used, however, cost functions were applied prioritizing diphone combi-

nations which belonged together, forming larger units. This way the non-uniformity was intrinsic.

Diphone based systems are almost out-of date, and probably are of historic importance only. Unit-selection on contrary is still in use, often as reference systems. In the Blizzard Challenge 2017 a hybrid system with a unit-selection mechanism at the backend won using an expressive speech corpus (please refer to Section 2.5 for more information).

In expressive speech synthesis, in order to produce expressive speech or voices, there are two options for concatenative systems. One, to record a new voice with the desired expressive speech style or emotion. Two, to manipulate the synthesized signal. In Section 2.5 some examples of concatenative and unit-selection expressive speech synthesis systems are given.

2.3 Statistical Speech Synthesis

In this work, statistical speech synthesis refers to *Hidden Markov Model (HMM)* based speech synthesis. This model is used to learn from extracted speech characteristics and to reproduce their averages and variations for given phonemes and contexts. Once the model reproduces the learned speech features, a filter is applied to convert them into a waveform.

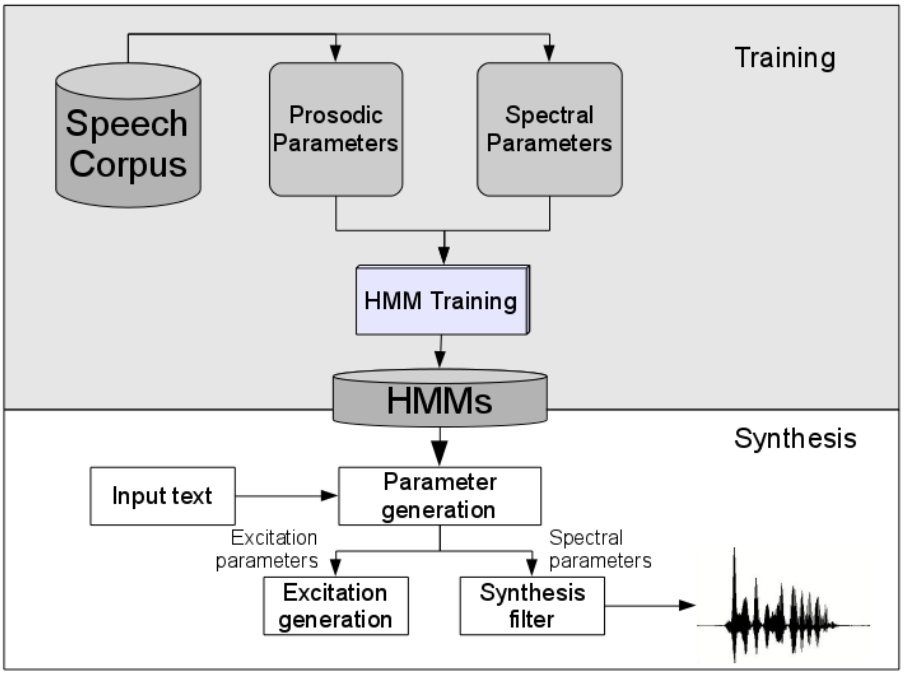
A Hidden Markov Model has N hidden states. The model begins in the state i with an initial probability π_i. Then, it changes to another state (or stays in the same) with a transition probability a_{ij}. From each state, the model emits a symbol with an emission probability b_{jk}. If A is the matrix of all transition probabilities, B is the matrix of all emission probabilities, and π is the vector of all initial probabilities, then an HMM is defined as:

$$\lambda = (A, B, \pi) \tag{2.2}$$

Usually in speech, unidirectional three- or five-state phoneme-HMMs are used, where each state represents the beginning, the middle, and the end of each phoneme. Hidden Markov Models had a long tradition in speech recognition. The first systems used discrete density HMMs, like by Baker (1975), and later continuous density HMMs, like by Juang (1984). An extensive introduction to speech recognition and the applications of HMMs in it can be found for instance in Rabiner and Juang (1993). There were early attempts to use HMMs in speech synthesis, such as by Ljolje and Fallside (1986), where HMMs were used for pitch contour generation, or by Giustiniani and Pierucci (1991), where the most probable acoustic feature vector is generated from a phonetic symbol using a phonetic and an acoustic HMM. However, the first successful system was developed by Masuko et al. (1996) introducing dynamic features in HMM based synthesis. The basic structure of the model is: perform MFCC analysis and prosodic analysis on a speech database and train phoneme based HMMs using the MFCCs and F0, and the derived dynamic features. In the synthesis part, an input text is converted into a phoneme sequence, and for each phoneme the corresponding HMM generates an MFCC and an F0 sequence. This sequence is then used for waveform generation with the *Mel Log Spectral Approximation*

(MLSA) filter, as in Imai (1983). Figure 2.8 shows the general architecture of an HMM based synthesis system.

Figure 2.8: HMM based system architecture.



The phoneme HMMs are first pretrained as monophone HMMs, and then reestimated as triphone models for all available triphones in corpus. The state durations are modeled by aligning the training data to the models via the Viterbi algorithm, obtaining sate duration density, each as a single Gaussian distribution. At synthesis time, q is the state sequence and o is the output parameter sequence, and it is obtained maximizing $P(q, o|\lambda, T)$, where λ is the HMM and T is the number of frames. If no dynamic features are taken into account, $P(q, o|\lambda, T)$ is maximized when $o = \mu$, generating discontinuities especially in the transition part of the phones. Including dynamic features, the means of the dynamic features are close to 0 at the static part, but high at the transition part of the phones, generating smooth parameters. To model coarticulation for non-existing triphones in the database, HMM states are clustered depending on the context. The system presented by Masuko et al. (1996) was implemented as the *Toolkit for HMM-based speech synthesis (HTS)* by Tokuda et al. (2008).

In comparison to concatenative speech synthesis, HMM-based speech synthesis is much more flexible and has little memory requirements. The disadvantage is the same as in other parametric speech synthesizers: low naturalness of synthetic speech. A partial answer to this problem is the usage of high quality vocoders like *Straight*, decomposing the speech signal into source and spectral parameters, applying refinements on each parameter extraction, by Kawahara et al. (1999), or *Ahocoder*, based on the *harmonic plus noise (HNM)* models, by

Erro et al. (2011b,a). In this work, Ahocoder is used for synthesis in experiments in Chapters 3 and 4, and Straight in Chapter 5.

The HNM model splits the spectrum in two parts, a harmonic (=voiced) lower band, and a noise (=unvoiced) upper band, as in:

$$s(t) = s_n(t) + s_h(t) \tag{2.3}$$

where the harmonic part $s_h(t)$ is defined as:

$$s_h(t) = \sum_{k=1}^{K}(t)A_k(t)\cos(2\pi k f_0 + \phi_k(t)) \tag{2.4}$$

where, $A_k(t)$ is the amplitude of kth harmonic at time t, and ϕ is the phase of kth harmonic at time t. The unvoiced part is modeled as filtered stochastic noise. Erro et al. (2011a) also proposes the usage of *maximum voiced frequency (MVF)* as an additional parameter in HNM-based vocoders. The MVF is the frequency which splits the spectrum into the harmonic and the noise part.

The flexibility of HMM-based speech synthesis has opened many possibilities for expressive speech synthesis. Similar to concatenative synthesis, a system can be trained using an expressive or emotional corpus. Also, model parameters can be manipulated as to control expressiveness in several ways. Some HMM-based expressive speech synthesis systems are presented in Section 2.5.

With all the advantages of HMM-based synthesis there is still one problem to solve, that is the problem of data insufficiency. Especially in expressive speech synthesis, where expressiveness can vary in unlimited ways, it is difficult to find enough training data to create robust voices with statistical models. This problem is partly solved using *speaker adaptive training (SAT)* described in Section 2.3.1.

2.3.1 Speaker Adaptation

Speaker adaptation comes from speech recognition and refers to modifying model parameters or feature vectors in order to recognize speech spoken by new speakers, according to Ortega and Miguel (2014). Model parameter adaptation modifies previously trained models in order to match test data. Some techniques for example are: *Maximum a posterior (MAP)*, where the posterior probability of the HMM parameters is maximized; *maximum likelihood linear regression (MLLR)*, where linear transformations of mean vectors and, possibly, covariance matrices, is performed to match the test data; *Vector Taylor Series*, where the nonlinearity between speech, noise and convolutional degradation is approximated with Taylor series. Feature vector based adaptation modifies feature vectors of test data in order to match the trained models. According to Ortega and Miguel (2014), some feature vector based techniques are: *Mean and variance normalization* and *multi-environment model-based histogram normalization (MEMLIN)*, where the input feature vectors are normalized towards the training data; *vocal tract length normalization* and *augmented state space acoustic decoder* try to model the differences between the trained models and the testing data focusing on the effects of the physiological differences between speakers.

Adaptation techniques are used in speech recognition in order to match test data of new speakers when little training data is available for these speakers. In speech synthesis, speaker adaptation is performed for the same reason, but with a different objective. The idea is to adapt a well trained speaker model with sparsely available data of new speakers in order to obtain new voices. In practice, the adapted speaker model is trained from multiple speakers such that a lot of variation is included in training data, making it easier to adapt the *average voice model (AVM)*. In expressive speech synthesis speaker adaptation techniques are very practical since in "real-life" corpora, like audiobooks, radio broadcasts, etc., most expressive styles are very sparse and there is not enough data to train all desirable voices. The most used adaptation techniques in speech synthesis are *Maximum a Posterior (MAP)* and *Maximum Likelihood Linear Regression (MLLR)*, and are described below.

Maximum a Posteriori (MAP)

Maximum a Posterior (MAP) tries to maximize the posterior probability of HMM parameters a posteriori, according to adaptation data. The parameters are updated maximizing:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}(P(\Theta|O,H)) = \underset{\Theta}{\operatorname{argmax}}(P(O|H,\Theta)P(\Theta)) \tag{2.5}$$

where $P(\Theta)$ is estimated from existing parameters, O is the new observation vector, H is the observation transcription and $\hat{\Theta}$ are the new parameters. Usually, only mean vectors are re-estimated, however also covariances and weights can be adapted. The estimation of the mean vectors is defined as:

$$\hat{\mu}_{k,c_k} = \frac{\tau \cdot \tilde{\mu}_{k,c_k} + \sum_{t=1}^{T} \gamma_{k,c_k}(t) o(t)}{\tau + \sum_{t=1}^{T} \gamma_{k,c_k}(t)} \tag{2.6}$$

where $\hat{\mu}_{k,c_k}$ is the re-estimated mean, $\tilde{\mu}_{k,c_k}$ is the prior mean, and γ_{k,c_k} is the posterior occupancy probability of component c_k in k-th state. $o(t)$ is the new data point at time t. τ is a meta-parameter which is used to bias the influences of the new and the prior data, usually set heuristically between 2 and 20.

The problem in MAP adaptation is that only model parameters of phonemes seen in adaptation data will change. However, usually, there are always unobserved parameters since the adaptation data amount is limited. *Regression based model prediction (RMP)* and *Structural MAP* are MAP extensions which try to deal with this problem, according to Ortega and Miguel (2014). RMP searches for correlations between parameters and tries to adapt not observed parameters using the linearly correlated observed ones. Structural MAP organizes the Gaussians of the acoustic model in a tree structure, where similar parameters are grouped on the same level.

Maximum Likelihood Linear Regression (MLLR) and Speaker Adaptive Training (SAT)

Maximum likelihood linear regression creates clusters of similar parameters and performs a linear transformation using regression matrices. The clustering is

performed statistically or using prior knowledge like phoneme's articulation type. The individual speaker characteristics are modeled projecting the speaker independent vector of means μ_j on the speaker dependent vector of means $\mu_j^{(r)}$, where r is the speaker, as in:

$$\mu_j^{(r)} = A^{(r)}\mu_j + \beta^{(r)} \tag{2.7}$$

where $A^{(r)}$ is the transformation matrix and $\beta^{(r)}$ is an additional vector with speaker specific information.

Anastasakos et al. (1996) proposes *speaker adaptive training (SAT)* used with MLLR, a technique used to maximize the likelihood of training data given MLLR adapted models. This techniques separates phonetic variations from the speaker specific variations using two criteria: A speaker independent component, and a speaker dependent component which acts like a filter. In a normal speaker adaptation (non-SAT) the optimal speaker independent HMM parameter set $\tilde{\lambda}$ is estimated as

$$\tilde{\lambda} = \arg\max_{\lambda} \prod_{r=1}^{R} \mathcal{L}(O^{(r)}|\lambda) \tag{2.8}$$

In SAT, the specific speaker transformations set $\tilde{G} = \tilde{G}^{(1)}, ..., \tilde{G}^{(R)}$ is added, while $G(r) = A^r, \beta^r$, and the HMM model parameter set $\tilde{\lambda}$ is jointly estimated with $\tilde{G}$ maximizing the likelihood of the training data as in:

$$(\tilde{\lambda}, \tilde{G}) = \arg\max_{(\lambda, G)} \prod_{r=1}^{R} \mathcal{L}(O^{(r)}; G^{(r)}, \lambda) \tag{2.9}$$

where $\mathcal{L}$ is the likelihood, $O^{(r)}$ is the speaker observations sequence of the speaker r, and R is the total number of speakers. $\tilde{\lambda}$ models only phonetic variations, not the speaker specific ones. The means are re-estimated as follows:

$$\begin{aligned} \tilde{\mu}_k = \Big\{ \sum_{r,t}^{R,T_r} \gamma_k^{(r)}(t) \;\; \tilde{A}^{(r)T} \; \Sigma_k^{-1} \;\; \tilde{A}^{(r)} \Big\}^{-1} \times \\ \Big\{ \sum_{r,t}^{R,T_r} \gamma_k^{(r)}(t) \;\; \tilde{A}^{(r)T} \;\; \Sigma_k^{-1}(o^{(r)}(t) - \tilde{\beta}^{(r)}) \Big\} \end{aligned} \tag{2.10}$$

where $\tilde{\mu}_k$ is the re-estimated mean of k-th Gaussian density, Σ_k is the covariance matrix, $o^{(r)}(t)$ is the t-th observation generated by r-th speaker, $\gamma_k^{(r)}(t)$ is the posterior probability that $o^{(r)}(t)$ belongs to the k-th Gaussian density.

Also here, the mean vectors are adapted, assuming that the variability is already trained in the *average voice model (AVM)*. However, additionally, covariances can be re-estimated, with the techniques *Variance MLLR* and *Constrained MLLR* (in the latter one the same matrix is used to adapt means and variances in order to reduce computational cost). In proper experiments, it was found

that the re-estimation of covariance matrices has a limited effect on the resulting synthetic speech. Sometimes it did affect the voice significantly, however, more often it had little effect.

As mentioned above, speaker adaptation is performed on a pre-trained *average voice model (AVM)*. The average voice model can be trained treating all speakers as one, however a better approach is proposed by Yamagishi (2012). Here, first the context and speaker dependent models are trained separately. Then, a shared decision tree is constructed from the speaker dependent models. The Gaussian probability distribution function for the average model is calculated combining all Gaussians of the nodes of the tree. Finally, AVM parameters are re-estimated with the SAT technique with the training data of all speakers, and state durations are obtained. The clustering procedure is also applied on the state duration distributions.

An example of practical application of SAT is *ZureTTS*, which was expanded in the framework of *eNTERFACE 14* at the beginning of this thesis, and gave an impulse to use speaker adaptation techniques for the present proposal. ZureTTS is a speech synthesizer which can be used to create personalized voices, aiming at persons with speech impairments. Available for several languages, users are asked to read a set of sentences, which are recorded and used for adaptive training. Further details can be found in Erro et al. (2014, 2015). Experiments on expressive speech synthesis with adaptation techniques are described in Chapters 3 and 4, where expressive voices are built via adaptation from unsupervised clusters of expressive speech.

2.4 Deep Learning

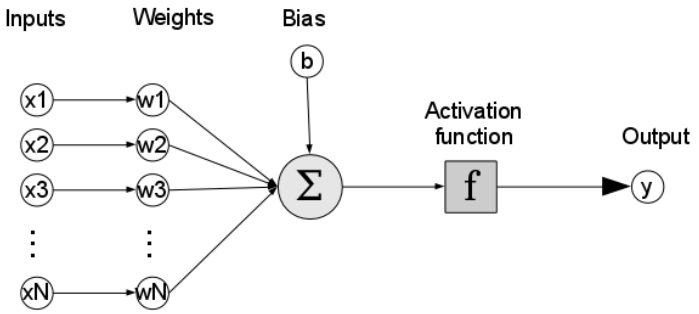
Deep learning refers to a specific kind of machine learning which uses *artificial neural networks (ANN)* to learn complex data patterns. Artificial neural networks, or simply neural networks (NN) were developed in the late 80th, and, as the name suggests, are inspired in biological neural networks. ANNs have been applied in many fields, however, only recently, computing power has advanced so much as to be able to train large and deep networks in reasonable time, specifically using GPUs. This development naturally caused the application of NNs in speech synthesis, with large success within about one or two years, almost instantly turning other synthesis techniques obsolete. Nevertheless, due to the recency of the "NN-boom", there have not been many applications yet for expressive speech synthesis, a fact that surely will change soon. While this thesis began, the NNs were practically never applied in speech synthesis, at least not for the synthesis itself. Now the experiments presented in Chapter 5 will be some of the pioneer experiments applying DNN-based synthesis with semantic sentiment vectors on expressive speech.

2.4.1 Introduction to deep learning

For the reason of the novelty of neural networks in expressive speech synthesis, a rather detailed introduction will be given on the neural networks itself, following expositions in Goodfellow et al. (2016), where more information can be found.

The basic unit of a neural network is the *neuron*, also called *perceptron*, and it is designed as illustrated in figure 2.9.

Figure 2.9: Artificial neuron.



Each value x_i of an input vector is multiplied by a weight w_i and they are summed up with a bias term b, as in:

$$a = \sum_i w_i x_i + b \tag{2.11}$$

then, an activation function is applied, as in:

$$y = f(a) \tag{2.12}$$

The activation function is usually a differentiable threshold function, such as the *sigmoid* function or *hyperbolic tangent* functions. The geometrical interpretation of the neuron is that of a hyperplane, where the weights control the rotation and the bias controls its position regarding the origin. The activation function performs a sort of a binary classification.

A single hyperplane is not enough to work with complex patterns, so the model is first extended introducing sets of parallel neurons. If the output y is interpreted as a probability $P(y = 1|x)$, then the output activation for the parallel neuron set can be defined as:

$$P(y = k|x) = \frac{e^{x^T w_k}}{\sum_{n=1}^{N} e^{x^T w_n}} \tag{2.13}$$

where k is the class, and N is the total number of neurons. This type of architecture allows to perform an N to M mapping. It is visualized in figure 2.10. Conventionally, the input values are considered to be the first layer, i.e. the input layer. The output layer is composed of M neurons, each of them "firing" an output y_i.

Figure 2.10: Neural network with 1 layer of parallel neurons.

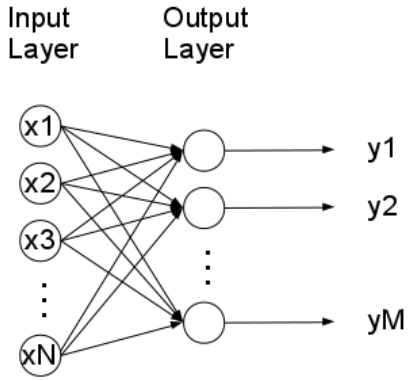
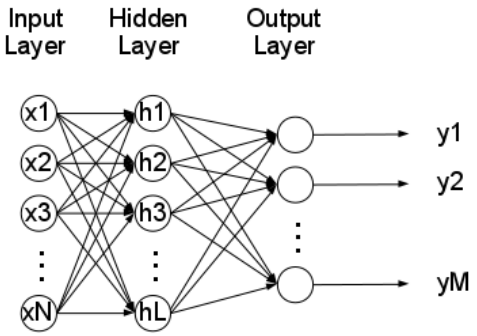


Figure 2.11: Neural network with 1 hidden layer.



To address more complex problems, one or more additional layers of neurons are introduced, called *hidden layers*, as illustrated in figure 2.11.

Here, the input layer is connected to the hidden layer, and the hidden layer's output is the input to the output layer, which, again, fires M outputs y. More hidden layers can be added to create deeper networks, where the output of each layer becomes the input of the next one. This forward directed architecture is called *feed-forward* architecture. The weight vector w can be rewritten as a weight matrix W_{ij}, where the weights serve for linking of the i-th and the j-th layer. Networks with many layers are called *deep neural networks (DNNs)*.

The standard training procedure of neural networks is called *back-propagation* and is generally defined in two steps. First, the error of the network output is calculated and back-propagated from the last to the first layer. Second, the weights of each neuron are updated. In order to define the back-propagation algorithm, the activation a is defined as:

$$a_j = \sum_i w_{ji} z_i \tag{2.14}$$

where z_i is the output of the neurons in the previous layer, which is the input to the current one, and w_{ji} is the weight from j to i. When the activation function is applied, the output is defined as:

$$z_j = f(a_j) \tag{2.15}$$

The error for each input/output combination t is defined as:

$$E = \sum_t E^t \tag{2.16}$$

where E^t is a differentiable function for each output as:

$$E^t = E^t(y_1 ... y_N) \tag{2.17}$$

Then, two error terms are defined. The error term δ_n for the actual network output values y_n and t-th output example is defined as:

$$\delta_n = \frac{\partial E^t}{\partial a_n} = g'(a_n) \frac{\partial E^t}{\partial y_n} \tag{2.18}$$

where g is the denomination of the activation function f for the output layer and g' is its first derivate. The error term δ_j is defined for the hidden units of the layer j as:

$$\delta_j = f'(a_j) \sum_n w_{nj} \delta_n \tag{2.19}$$

Each error term δ_j is obtained from the neurons of a posterior layer δ_n, where the error term for the output layer is computed first. This way the error back-propagates to the first layers. The next part is weight updating. A common method is the so-called fixed-step gradient descent learning where small parts of the weights are subtracted depending on the error term, as in:

$$w_{ji}^{(k+1)} = w_{ji}^{(k)} - \Delta w_{ji}^{(k)} = w_{ji}^{(k)} - \eta_0 \sum_t \delta_j^t z_i^t \tag{2.20}$$

where (k) is the current iteration and η_0 is the *learning rate*. Learning rate is an important parameter which has to be set heuristically beforehand. If the learning rate is too high, the learning will diverge and the error will increase. If it is too low the network can get "stuck" in local minima.

Other terms of gradient descent techniques are *batch gradient descent*, where the gradient of the cost functions is updated for the whole training set; *stochastic gradient descent (SGD)*, where a parameter update is performed for each training example; *mini-batch gradient descent*, where an update is performed for every n training examples.

The gradient descent techniques sometimes suffer problems with choosing the correct learning rate, applying the same learning rate for different parameter frequencies, different steepness in different dimensions, etc. Therefore, gradient optimization techniques exist, which try to solve these problems. Some of the most important ones are listed below. *Momentum*, by Qian (1999), tries to prevent SGD oscillating due to different steepness in different dimensions and bring it faster to the local optimum; *Nesterov accelerated gradient*, by Nesterov (1983), where the gradient is calculated not with respect to the current parameters, but with respect to the approximated future position of the parameters; *adaptive gradient algorithm (Adagrad)*, by Duchi and Hazan (2011), gradient-based optimization, i.e. learning rate is adapted to the parameters, storing past square gradients, while less frequent parameters underlie larger updates and more frequent parameters, smaller updates; *Adadelta*, by Zeiler (2012), windowing square gradient storage; or the similar method *RMSProp*, by Tielemann and Hinton (2012); *adaptive moment estimation, (Adam)*, by Kingma and Ba (2014), which computes learning rates for different parameters from estimates of the first (mean) and the second (uncentered variance) moment of the gradients.

Another problem with learning is the *overfitting*. Overfitting means that the network adapts to the noise in the training data failing to generalize. In order to avoid it, a common technique is the *dropout*. Dropout means that the connections of individual units are temporarily deactivated in order to prevent them to adapt too much to the data in each iteration.

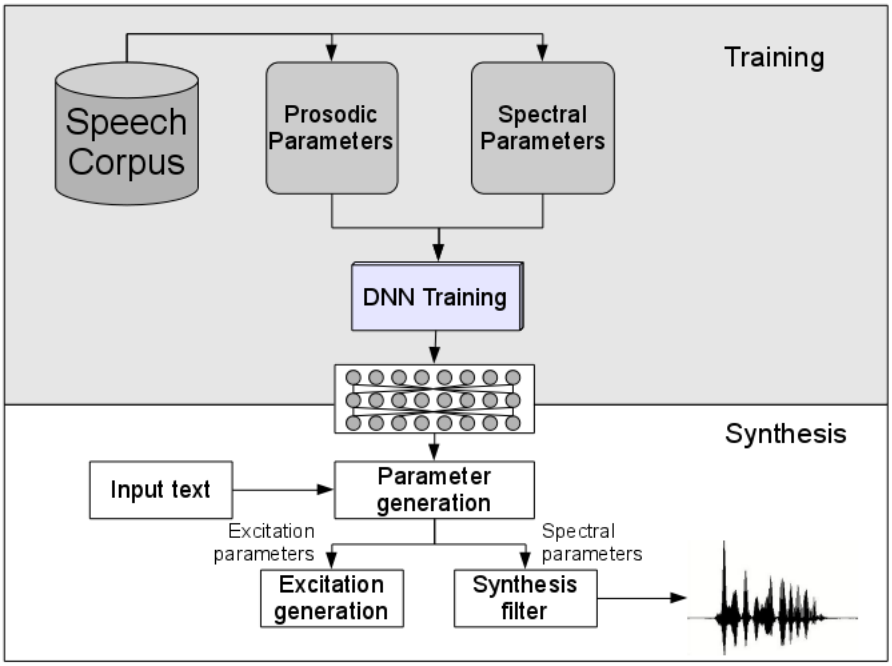
In order to process temporal sequences it is useful to introduce a "memory" feature into the networks, such as in the *recurrent neural network (RNN)*. The main characteristic of RNNs is that each neuron not only receives the input at time t, but also a memory state h_{t-1} at time $t-1$. These networks model well short time dependencies, however for longer time windows they face problems with vanishing or exploding gradients since these are multiplied at each time step. An improvement to RNNs is for instance the *long short term memory (LSTM) cells*. These units introduce the so-called gates which decide, which information from the input or the past state enters, leaves or is deleted (input, output and forget gates). Networks using the LSTM cells are capable of modeling long-term time dependencies very well, however, they have a large number of parameters compared to the RNNs, so the training is much more complex. Other alternative approaches to LSTM are for instance *Phased LSTM*, *Gated Recurrent Units (GRUs)*, and other versions with modified gates or algorithms.

2.4.2 Neural Network Based Speech Synthesis

Zen et al. (2013) proposed a parametric synthesis system based on DNNs. The system's architecture is derived in a relatively straightforward manner from the architecture of HMM-based synthesis, practically replacing the HMMs by the DNNs, as shown in figure 2.12.

The advantage in the DNN-based synthesis lies in the fact that the network weights are trained using all data, i.e. there is no clustering and no information loss. Also, DNNs are rather capable of learning complex relationships in the input data, in comparison to the HMMs and decision trees. Trajectory smoothing is also learned by the DNNs. The results achieved by Zen et al. showed that

Figure 2.12: DNN based system architecture.



four-layer deep DNN-based systems significantly outperformed HMM-based systems, using the same waveform generation method. Among others, DNN-based approaches are used by Qian et al. (2014); Hu et al. (2015); Valentini-Botinhao et al. (2015); Zen and Senior (2014), or Lu et al. (2013), this latter one adding semantic vector representations as a linguistic input feature.

Other types of NN-based architectures were proposed, for instance Wu et al. (2015b) apply a bottleneck DNN design, i.e. a hidden layer with a relatively smaller amount of neurons than the prior and the posterior layers, for a compact representation, achieving a better modeling of frame interdependencies. RNN and LSTM have been used in speech synthesis by, for instance Chen et al. (1998); Fernandez et al. (2014); Achanta et al. (2015); Zen and Sak (2015); Wu and King (2016), among others. Using LSTMs the temporal dependencies are captured intrinsically and there is no need to apply smoothing algorithms.

Also, speaker adaptation has been performed with neural networks, for instance, Wu et al. (2015a) compare different approaches, i.e. adding identity information to the input features, output feature space transformations, and *Learning Hidden Unit Contribution*. Pascual and Bonafonte (2016a) perform interpolation between speakers in a RNN-LSTM framework. A similar approach called *cluster adaptive training (CAT)* is presented by Tan et al. (2016).

A new direction in synthesis with neural networks is the generation of raw waveforms instead of speech parameters. The reason is, first, that speech parameters, no matter how good they are, will not perfectly represent the waveform with

all the nuances. Second, speech parameters need a filter model which converts them back to the waveform. Basically, there is a necessary information loss and adding in parameterizing which contributes to lower quality. *WaveNet*, by van den Oord et al. (2016), is probably the most famous system so far which generates raw waveforms sample by sample, though being computationally very expensive, taking minutes and hours to synthesize few seconds of speech. Arik et al. (2017) presented the system *Baidu*, achieving real-time synthesis. Finally, Wang et al. (2017) have presented the system *Tacotron*, which is an almost end-to-end synthesis system, where audio is learned directly from $< text : audio >$ pairs, achieving very natural voice quality. The system learns from text characters, however, the output are spectral frames, which are then converted into waveform.

In general words, the NN-based speech synthesis develops and changes incredibly fast. Before little more than half a year WaveNet appeared, considered to outperform (almost) everything else in quality, and only few months later two systems have been developed which outperform WaveNet, at least on the engineering level. Probably, this review will loose its currency in a few months, and also expressive speech synthesis will completely change.

2.5 Expressive Speech Synthesis

After the introduction to common speech synthesis methods and architectures, and before continuing with expressive speech synthesis, first a very general question will be addressed: *What is expressive speech?* The Oxford dictionary defines the word "expressive" as something that is *Effectively conveying thought or feeling.* According to this definition, "feeling" is kind of a part of expressive speech. In fact, in numerous occasions, in the literature or speech technology community, expressive speech is treated as equal to emotional speech. However, also conveying of "thought" is part of the definition, which means that also other type of information is part of expressiveness. For instance, one could think that pragmatic stress, focus, or speaking styles, like for instance politician speech, or news, all that is expressive, but not necessary emotional.

Regarding emotion, Ekman (1972) defines six emotional categories which have been used in a standardized way in speech sciences: *disgust, anger, fear, joy, sadness* and *surprise.* This taxonomy is not unproblematic. First, it is not clear whether it is complete. Second, it does not reflect graduation of emotions or mixed emotions. For instance, one can be very angry, a little angry, and so on. Also, although the label *angry* can be put on emotional realization of anger, different people will realize it in very different manners. Applied to speech, this has a direct effect on acoustics. On the other hand, for instance *surprise* can be mixed with other emotions yielding very different expressions, like for example {*positive, negative, angry, happy, fearful*} surprise.

Schlossberg (1954) proposes a different approach to classify emotions, describing them in three dimensions: *activation*, *evaluation*, and *power.* This system has been further developed by Kehrein (2002), proposing the three dimensions *activation*, *valence*, and *dominance.* Activation is the excitation of an emotion, valence is the positiveness, and dominance is the strength. These three primitives are related to prosodic features of speech, such as *speech rate*, *F0 range*

and curves, etc. In some works, as by Schuller (2000); Lopez-Otero et al. (2014), the three-dimensional model has been used in emotion recognition.

Actually, it is not even really clear, what emotions are. Many perspectives define them differently, like philosophical, psychological, evolutionary, cultural, etc. A more complete review on what emotion is can be found in for instance Cowie et al. (2001).

Generally, opposed to "expressive speech", the term "neutral" is common in use. However, the definition of "neutral" is not very clear, and sometimes might even not make sense. Often the question is asked, if *the synthesized speech is expressive or not (neutral)?* Here, "neutral" or "not expressive" speech is often everything which is not "happy", "sad", "angry", and so on, i.e. basically something which does not belong to a limited number of categories and is somehow in the middle of everything. It could be seen as a kind of origin in a expressive space, while the expressions would represent some points more or less distanced from that origin. In this continuous case, it is not really clear, until when neutral speech is neutral and when it starts being something else. Contrary on these "classifying" tendencies, what is being proposed in this work is, that dealing with expressive speech, basically everything is expressive, even the "neutral" speech. What rather matters is, *if the expressiveness of speech is adequate for the current situation.* Therefore, the definition of expressive speech which this work follows is rather *adequate speech*, in the sense of speaking style, stress, focus, emotion, and so on. Related to this definition is the question of the acoustic realization of expressiveness. Generally, speech is represented in terms of meaningful parameters, which, in speech synthesis, are analyzed and transformed back to speech. If this is true, then, expressiveness, as being part of speech, should also be able to be described in terms of parameters or features. This problem is not trivial and will be addressed with more detail in Chapter 3.

Transporting the above definition to the speech synthesis problem, there are two main problems which need to be dealt with. The first problem which needs to be solved, is to provide the machine the capability to generate expressive speech. The information about the expressive style can be fed into the synthesis system as an additional input feature, given *which* expression to synthesize. Furthermore, how this information is processed depends very much on the system architecture, but also on the intended approach. Basically, there are three major ways of synthesize emotional speech. Govind and Mahadeva Prasanna (2013) call them by "explicit control", by "playback control", and by "implicit control".

Explicit control means modifying neutral synthesized speech by applying mainly prosodic rules, modifying synthesis parameters or via post-processing with algorithms like PSOLA. This type of approaches reflects the tradition of assuming that prosodic features are, if not exhausting, but enough to synthesize emotional speech, as for instance states Vroomen et al. (1993), or Schröder (2009), whereas the latter one also addresses voice quality. The explicit control paradigm has been applied to many types of TTS. For instance, to formant synthesizers by Cahn (1989); Murray and Arnott (1993, 1995); Burckhardt and Sendelmeier (2000), since it is fairly easy to modify the parameters of formant synthesizers, as for instance argues Schröder (2001). In diphone synthesis, for instance

Vroomen et al. (1993); Montero et al. (1999), the first one applying PSOLA, and the second one so-called *copy-synthesis*, where individual diphones were copied from different emotional databases regarding prosodic requirements. Also, in this latter work, Montero et al. (1999) found that some emotions can not be modeled by prosodic means exclusively, like for instance *cold anger*. Carbal and Oliveira (2006) propose a system which performs prosodic transformations and changes to glottal source parameters, and can be applied to synthetic and natural speech.

Playback control refers to speech synthesized directly from a corpus, by means of unit selection or trained statistical models. In unit selection, some systems to mention are those developed by Iida et al. (2000) and Campbell (2006), or the IBM expressive speech synthesis system by Hamza et al. (2004), this latter one designed for reading positive and negative news, technically combining a purely corpus-driven approach with an explicit modeling of prosody. On the HMM side, Yamagishi et al. (2003) modeled several emotions and speaking styles individually. Yamagishi et al. (2005) compared this approach to another one, where only one model for all speaking styles was trained, and the speaking styles were treated as context, with similar results. In general, the problem of the playback control is the need for a large amount of data, especially for unit selection.

Implicit control basically means interpolation between two or more models. Here, especially the statistical systems are in question, where with techniques like SAT or voice transformation different expressive styles can be trained with relatively small amount of data. Some examples of HMM-based expressive synthesis were developed by Tachibana et al. (2005, 2006); Nose et al. (2005, 2009). Also, speaking style *transplantation*, i.e. the transplantation of one speaking style to a different person's voice has been investigated by for instance Lorenzo-Trueba et al. (2013, 2014); Lorenzo Trueba (2016). Also, in singing voice synthesis, which could be understood as a kind of a special case of expressive speech synthesis, HMMs have been used, by for example Saino et al. (2006).

On the neural network side, less has been done related to expressive speech due to the novelty of the models in speech synthesis. Here, an early approach is proposed by Sato and Morishima (1996), where an emotional space is modeled and used to modify timing, pitch and intensity of synthesized speech. Generally, NN-based speech synthesis can be used in a similar way for expressive synthesis as the HMM-based one. In chapter 5 some experiments related to NN-based synthesis will be presented and discussed. In `http://emosamples.syntheticspeech.de/` many synthetic samples from different expressive speech synthesis systems are presented and can be compared.

Also, much effort on creating synthetic expressive speech has been invested in the *Blizzard Challenge*. Blizzard Challenge was first organized in 2005 and was a TTS challenge which first focused on naturalness and intelligibility. When TTS systems got better though, additional aspects were focused on such as specific domain tasks and expressive speech. In 2007 a paper was presented with a TTS system for an online role-playing game, requiring specific concatenative expressive speech synthesis, as in Rozak (2007). For the first time, in 2008 the English corpus used in the challenge contained expressive speech, although it was not the focus of the evaluation. In 2011, a methodology for audiobook

reading evaluation was proposed by Hinterleitner et al. (2011) and several German voices were evaluated. Their proposed methodology was further used in future challenges. The first time an expressive speech related task was explicitly included in the evaluation was in 2012, providing as training material four Librivox audiobooks recorded by a US speaker. However, only one participant took part in the evaluation of the audiobook reading. In 2013 and in 2016 audiobook databases were used for the challenges and the tasks aimed at book reading.

The system with the most wins in the Blizzard Challenge is the *USTC*, developed by Chen et al. (2016), won from 2006 to 2012, was best in several categories in 2015, and won in 2016. The first version of their system was a parametric HMM-based system, which then converted to a hybrid system which is HMM-based, but uses concatenation for waveform generation. The HMMs have been used to drive the system until 2016, where they were replaced by LSTM-networks which were used for everything, except for the waveform generation itself, which is still done via unit-selection. In 2013, also a hybrid HMM-Unit-Selection system, *SHRC-Ginkgo*, won the challenge, though competing with USTC. 2014 was focused on Indian languages and the winner system was a unit-selection called *ILSP/INNOETICS TTS*, developed by Chalamandaris et al. (2014). 2015 was also aimed at Indian languages and, among the USTC system, another standing out system was a unit selection system developed by Rallabandi et al. (2015).

The second problem with expressive synthesis, and maybe of higher importance for this work, is to know, *when* to use which type of expressive speech. Humans do that naturally, conveying pragmatically thoughts, feelings, and so on. So, the most straightforward way to deal with this first problem is to let humans choose how to synthesize something, like they choose how to say something. This works, but, when the amount of expressive styles rises, in applications like book reading, the choice becomes very complex and possibly unnatural, since the number of all possible expressive styles is practically infinite. Also, expressiveness is speaker dependent, so each expression might sound very different produced by different speakers. Another option to deal with this problem is to let the machine deduce expressiveness automatically from text. In order to do that an advanced text analysis is needed, not only by methods described in section 2.1.1, but one which needs to capture semantic, pragmatic, and contextual relations in a text.

Despite of the variety of methods and systems presented above, most of them do have one drawback, and this is that most of them require a manual model adjustment, or labeling, or recording of training data, i.e. manual work. Nowadays, huge amounts of data are available, especially on Internet. It is impractical to use manual approaches in order to deal with it. Also, in order to account for the practically infinite amount of expressive styles, automatized techniques must be applied to deal with data. The aforementioned problem of predicting expressiveness from text is addressed in for instance Chen et al. (2014); Jauk et al. (2016); Jauk and Bonafonte (2016a); Lorenzo Trueba (2016). The methods proposed in these works can also be used for data clustering in order to gain training data, as done for example by Watts (2012)). Also, unsupervised methods on acoustic level can be applied to generate data clusters which can be used for training, as for instance done by Eyben et al. (2012); Jauk et al. (2015); Jauk and Bonafonte (2016b). Chapter 3 will treat this topic, among others. NN-based approaches offer automatized possibilities to deal with data and/or derive expressiveness from text, avoiding clustering. However, some sort

of control mechanism is needed which tells the system which expressive style is due at each moment. Some first ideas and experiments in this are presented in Chapters 4 and 5.

2.6 Discussion

This chapter has provided a compact introduction to speech synthesis systems and a state-of-the-art review of expressive speech synthesis. General aspects and basic procedures of TTS systems, such as text analysis, prosody prediction and waveform generation, and system architectures based on unit selection, HMMs and NNs, were presented and discussed. Also, a short introduction on deep learning was provided. In the state-of-the-art review of expressive speech synthesis the focus was put on the different methods of how expressive speech synthesis can actually be implemented, and examples of real systems were presented for each of the approaches.

An important point in the TTS development of past two or three years have been neural networks, which experienced a large boom especially in the last few months, from *Speech Synthesis Workshop 9* on, where WaveNet was presented. The usage of neural networks has created an impulse towards fully NN-based end-to-end systems where synthesis is performed directly generating waveforms from text, which just before some years yet was considered practically impossible. This new paradigm will surely completely change the TTS panorama within proximate time, turning everything else obsolete. An interesting reflector of TTS systems is the Blizzard Challenge, where until now unit selection was always involved in the winning systems. Now, the coming Blizzard Challenges will probably experience the paradigm change.

It is valuable to express again, that synthesis techniques, which are considered obsolete in general TTS systems, are often still in use in expressive TTS. The reasons for this are, the simplicity and controllability of the more simple techniques, and therefore their usability in fundamental rather than technological research, and, lack of usable expressive data. Especially looking at the state-of-the-art neural network based TTS, large amounts of data are crucial for the training of these systems. To get large amounts of controlled expressive data considering the high variability is still very difficult.

In expressive speech synthesis (but not only), data handling is one of the main problems, especially regarding modern ways of communication, such as Internet. Large amounts of usable data are available, however, it is often not controlled by the developer, and there is lack of automatic unsupervised methods of classifying this data such that it can be used to efficiently train speech synthesis models. Also, the information about expressiveness which can be gained from plain text is another important area of research which needs to be exploited for future applications. This thesis addresses mainly these two problems, from the perspective of unsupervised and automatized learning and data treatment. This focus away from the actual synthesis method, which drives towards completely NN-based systems, keeps being up-to-date and can equally be applied to NN-based systems. Surely, insights from this work can help to develop state-of-the-art NN-based expressive speech synthesis.

The unsupervised data clustering in the acoustic domain and acoustic features for the representation of expressive speech will be addressed in Chapter 3. The prediction of expressiveness from semantics is the main topic in Chapter 4. Neural network based expressive speech synthesis is handled in Chapter 5.

Chapter 3

Feature Selection for Expressive Speech Synthesis

Expressive or emotional speech synthesis has generally used manually annotated data for model training. To label or to record expressive speech is a very laborious and expensive work though, impracticable on large amounts of data. Also, on the other hand, emotional labels are very difficult to define despite the traditional references to emotions like *anger*, *joy*, *sadness*, etc. This is because the acoustic realization of emotions and other expression types is highly dependent on speaker, situation, graduation of expressiveness and combinations of expressiveness. Further details on this are discussed in section 3.2. Nowadays, Internet is a huge source of data, also of expressive and emotional speech data. However, in order to be used efficiently, this data needs to be organized. Since manual labeling can not be efficiently applied, automatic, non-supervised methods are needed. Clustering of data is one such method. Clustering relies on features which meaningfully represent data, here in terms of expressiveness or emotions. Features on different levels can be taken into account in expressive speech, like acoustic, prosodic, and possibly, related linguistic features. All of them are meant to represent the speech signal such that useful information can be derived and processed, preferably automatically. However, all of them are multi-functional, such that different phonetic, linguistic and extra-linguistic functions, like phone identity, accent, modus, focus, language identification, speaker identification, expressiveness, etc., are carried by multiple features. At the same time each feature is always related to multiple functions.

Which features are good for representing expressive speech is not very clear and different options have been proposed so far, as discussed in section 3.1. To understand how theses features can be related to expressiveness, it is necessary to understand what exactly they represent and how are they derived. Sections 3.1.1 and 3.1.2 discuss the most commonly used features related to expressive speech. Section 3.1.3 explains *i-vectors* as proposed feature for multi-speaker expressive corpora and section 3.1.4 introduces the reference feature set *openSMILE*, by Eyben et al. (2013), widely used in current emotional analysis. The objective is to find an optimal feature set for expressive speech data for automatic, non-supervised processing. For this purpose, in section 3.2 several

experiments, which evaluate different features and feature sets and their corresponding results, are presented. Finally, the overall results are discussed in section 3.3.

3.1 Acoustic Features in Expressive Speech: An Overview

Acoustic speech signals are described in terms of a set of acoustic features, where each feature fulfills several functions, i.e. is part of a subset of features which describe a certain phonetic phenomenon, such that $K \subseteq N$, where N is the whole set of acoustic features and K is the subset which describes a certain phenomenon. The problem with this definition is that, generally, acoustic features are not totally independent, i.e. not orthogonal. So, in the most phenomena all features are involved, but with different degrees of importance.

In the case of phonetic segments, it is relatively easy to find combinations of acoustic features which allow a reasonable identification of phonetic segments. Expressiveness is clearly a suprasegmental feature, however it surely also has segmental manifestations. For example, the configuration of the vocal tract for *screaming* as an expression of *anger* surely will affect spectral features on segmental level through wider opening of the mouth.

A different problem is the expressiveness by itself, or rather its acoustic realization. Continuing with the example of *anger*, the expression of anger can vary significantly between "type" of anger and speaker. For instance it can be expressed as aggressive screaming, controlled cold anger, hysterical, surprised, scared anger, and so on. All these expressions are *anger*, but very distinct acoustically. The differences can range from smaller variations to completely different expressions. Here, *anger* is only one example, the same problem applies to practically all expressions.

Traditionally, in expressive speech, prosodic features have been used for the representation. For instance, Kehrein (2002) used fundamental frequency, intensity and duration in his experiments, Szekely et al. (2011) used glottal source parameters to perform clustering of expressive speech styles in audiobooks. Pérez and Bonafonte (2005); Schuller et al. (2005) used a set of mainly prosody-based and some spectral based features for emotion recognition. Eyben et al. (2012) used prosodic features, i.e. F0, voicing probability, local jitter and shimmer, and *logarithmic HNR* for audiobook clustering and posterior synthetic voice training. In fact, Eyben et al. (2012) state, spectral features are considered to be poorly related to expressiveness, which is a general opinion on this topic (compare Vroomen et al. (1993); Schröder (2009)). However, some approaches showed that spectral features are also important for the discrimination of expressiveness. For instance, Barra-Chicote et al. (2010) suggest that different expressions are better characterized by different features; for instance, *anger* is rather characterized by spectral parameters, while *happiness* and *disgust* are better represented by both prosodic and spectral features. Also, Montero et al. (1999) finds that some expressions (emotions) are not well represented by prosodic features, e.g. *cold anger*.

Probably, dealing with real conversational speech, many factors such as multiple

speakers, multiple distinct realizations of the same expression, background noise, etc., must be taken into account. Recently, in speaker verification, *i-vectors* have proved to be an efficient feature, (e.g. Reynolds et al. (2000); Dehak et al. (2011)). In recent experiments by Lopez-Otero et al. (2014); Jauk et al. (2015); Jauk and Bonafonte (2016b), i-vectors have also been used in emotion prediction and expressive speech clustering. In fact, feature sets which contained i-vectors achieved the best results in multi speaker databases. On the other hand, traditional features performed better in single speaker laboratory speech databases. Eyben et al. (2013) propose a feature extractor, *openSMILE*, which is specialized to extract features for expressive speech, including thousands of possible features. Feature combinations based on this toolkit have widely been used in many expressive/emotional speech related tasks, for instance in the Blizzard Challenge. The following sections introduce spectral and prosodic features for expressive speech, as well as i-vectors and the openSMILE toolkit.

3.1.1 Spectral Features

Speech signal is produced through modifications of an excitation signal which can be stochastic, in form of noise, or periodic, produced through vibration of the *glottis (=vocal chords)*. The excitation signal is then modified in the vocal tract according to the acoustic theory of articulation (an introduction can be found in Vary et al. (1998)). Different articulatory configurations of the vocal tract cause variations in different frequency bands and allow the discrimination of phonetically meaningful speech realizations. Spectral features are obtained by converting the speech signal from time domain to the frequency domain, where energy in the different frequency bands becomes observable. The frequencies with the highest energy portions are the resonant frequencies and are called *formants*. Formants are frequencies, which wave longitude is such that, at the vocal tract ends pressure $p(t) = 0$ or rather the particle velocity $v(t) = 0$, also called *standing wave*, as described for instance by Vary et al. (1998) (figure 3.1). In reality, the values do not reach 0 and other variations occur due to the imperfectness of the vocal tract form, material, energy loss, etc. These imperfections and variations actually make human voice sound "human".

Due to narrowing and/or opening in determinant positions in the articulatory process, formant frequencies change, resulting in acoustic characteristics of a certain phonetic realization. Especially, vocal tract changes at other points with $p = 0$ or $v = 0$ yield acoustic changes and cause the co-articulation effects. In phonetic segmental analysis, mainly the first two or sometimes three formants are used. The wavelength for higher formants is so short that vocal tract modifications do not have much effect on it, so they are considered to be generally invariable and speaker specific.

Figure 3.2 shows the waveform and the spectrogram of a realization of the word Harry spoken in Spanish, [hari] in IPA, with formants drawn with red dotted lines. The formants are located at the frequency levels with highest energy. In the non-periodic regions of the realization of [h] no clear formants can be observed, however, it does show specific noise distribution.

One of the drawbacks of formants is that they are not clearly defined for non-vocalized sounds, such as fricatives. For this reason they are not robust enough

Figure 3.1: Standing waves of wavelength λ in a vocal tract of length l (from Vary et al. (1998)).

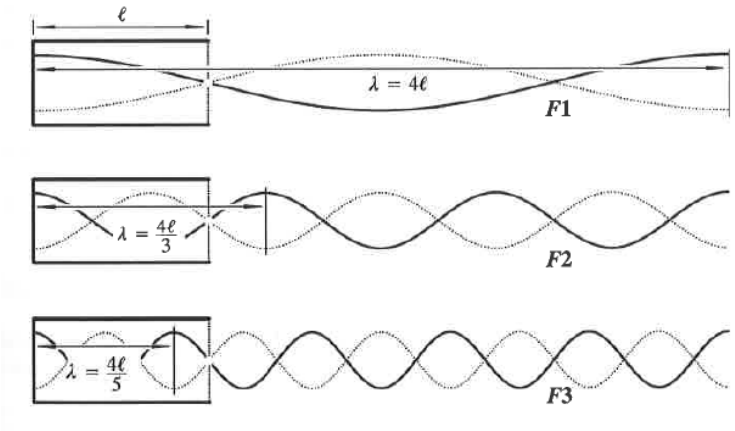
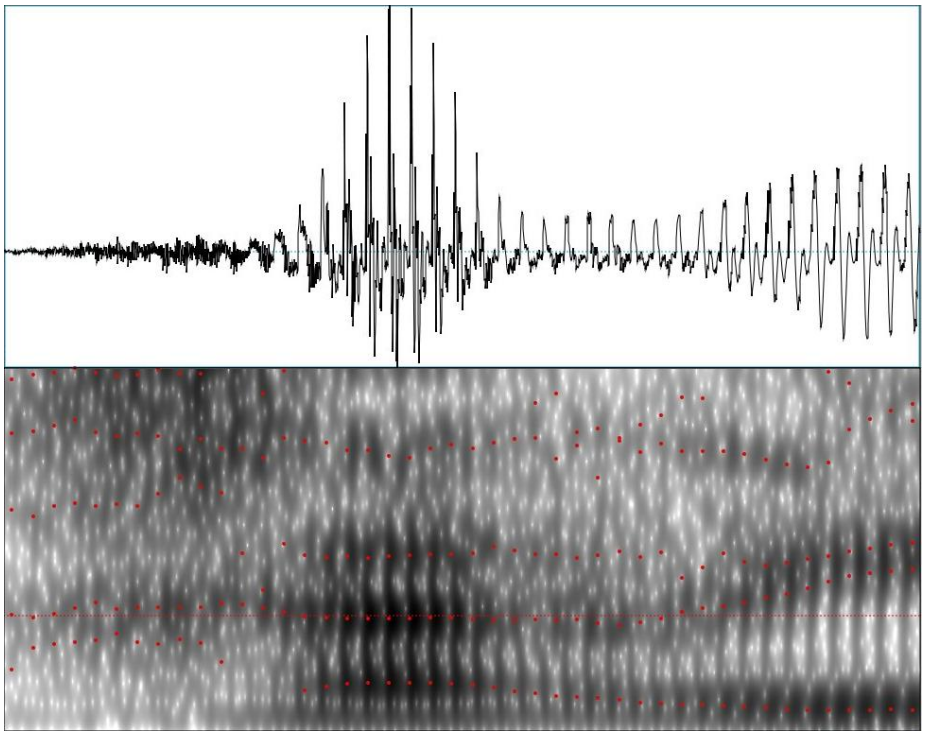


Figure 3.2: Waveform and spectrogram for the realization of [hari]. Formants are indicated by red dotted lines.



for automated speech processing. Instead, *Mel Cepstrum Coefficients (MFCCs)* have been very popular in all types of speech technology. MFCCs is a represen-

tation which is partly based on the human perception, the so called *Mel scale*, as by Stevens et al. (1937). The Mel scale is a scale of pitches which are perceived as equidistant. The MFCCs are calculated by, first, applying the *Fourier transform* on a windowed signal; second, applying the Mel-filterbank with M filters on the spectrum; third, calculating the logarithm of the resulting Mel frequency powers; and finally, performing the *Discrete Cosine Transform (DCT)* on the mel log powers. The resulting signal are the *Mel Frequency Cepstrum Coefficients*.

Generally, spectral features are considered to account more for segmental speech aspects than for suprasegmental. The specific vocal tract configurations yield the necessary resonance frequencies which determine the phone identity. Expressiveness is considered to be suprasegmental, nevertheless, certain emotions or expressions can affect the vocal tract configuration in a substantial way such that resonance frequencies change due to the expressiveness, not to the segment. Such findings have been made for instance by Barra-Chicote et al. (2010) or Montero et al. (1999).

3.1.2 Prosodic Features

Prosody in linguistics includes the description of three terms, as argued for instance by Vary et al. (1998): *quantity*, *intensity* and *intonation*. Prosodic features are widely considered to be suprasegmental and mainly responsible for the acoustic sound impression. In written language, there are only few clues of how to interpret the text prosodically, which basically can be resumed in punctuation. The three aforementioned concepts can be explained as follows:

- *Quantity* is everything related to the time structure of speech, such as speech rhythm and tempo, segment durations, pauses, etc. The acoustic counterpart is the *(segment) duration*.
- *Intensity* is especially related to the loudness, but also to accentuations and stress, although not exclusively of course. The acoustic counterpart is the *amplitude*.
- *Intonation* is everything related to the melody. Intonation is probably the functionally most loaded feature since it is related to almost all other prosodic (and partly also non-prosodic) aspects of speech. It is directly involved in stressing, phrasing, focus, modus, etc., and also to extralinguistic information such as emotions, speaker identity, etc. The acoustic counterpart of intonation is the *fundamental frequency (F0)*.

The acoustic counterparts mentioned above are not always perfectly clear, such that the linguistic concepts and the acoustic counterparts are cross-related with different degrees of influence. Also, the functionality of the linguistic concepts and their counterparts, like accounting for stress, emotions, etc., is normally not exclusive, i.e. various concepts and their acoustic counterparts are involved in almost all functions, also with different degrees of influence. Often the linguistic functions are language dependent, so the functionality of these features is changed or limited. For instance, in German or Italian *duration* is a distinctive

feature, i.e. the duration of phonetic segments can determine the meaning of the word [1]. For this reason the usage of duration for stressing is relatively limited since it can lead to ambiguities in understanding, so mainly pitch is used for stress. On the other hand, Mandarin is a tonal language and pitch is a distinctive feature, so its function as stress carrier is limited in Mandarin, as well as in other tonal languages, according to for instance Vary et al. (1998).

Also, very popular in prosodic analysis, especially in clinical analysis and pathology detection, two additional measurements are used: *Jitter* and *Shimmer*. Jitter is the average difference between the duration of consecutive periods of the fundamental frequency. The absolute jitter is calculated as:

$$jitt_{abs} = \frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i+1}| \tag{3.1}$$

where T_i is the period length of the F0 and N is the total number of the periods used to average. Other versions of jitter is the *relative jitter*, where the absolute jitter is divided by the average period length; *relative average perturbation*, where the difference is calculated between the current period's jitter and the jitter of its four closest neighbors; among others.

Shimmer is the variability between the amplitudes of two consecutive periods of the fundamental frequency. The absolute shimmer (in decibels) is calculated as:

$$shimm_{abs} = \frac{1}{N-1} \sum_{i=1}^{N-1} |20 \log(\frac{A_{i+1}}{A_i})| \tag{3.2}$$

where A_i is the maximum amplitude in a period and N is the total number of periods. Also here, different versions of shimmer exist, such as relative shimmer (normalized by the average amplitude difference), or *ddp*, i.e. relative difference between two consecutive differences, etc.

Jitter and Shimmer are commonly used in pathological speech analysis, resulting very robust in detection of certain pathologies. Also, they have been introduced in speaker recognition, for instance by Farrús et al. (2007), and speaker diarization by Zewoudie et al. (2014). Reflecting aspects of voice quality, they are also considered to be useful for expressive speech, being part of the features sets extracted by openSMILE, which will be discussed in Sections 3.1.4 and 3.2.4. Experiments with feature sets including Jitter and Shimmer will be presented in Sections 3.2.3 and 3.2.4.

In general, as argued above, prosodic features are considered to be crucial for emotional speech. Among with the terms *quantity*, *intonation* and *intensity*, also *rhythm* is reported to be an important feature in description of certain speaking styles. Rhythm is a durational feature, however, it does not refer to the bare duration of segments or pauses, but rather to the temporal organization

[1] In German, the duration of the vowels is distinctive in a phonological and prescriptive sense. In phonetic reality, the tension of the vowels is more distinctive, where traditionally it is considered that tensed vowels are also lengthened, a discussion on this topic can be found for instance in Wiese (1988).

and structure of speech. Wagner (2008) reports that rhythm is crucial for speech understanding and information processing by humans, and is related to human perception of time. For expressive speech, speaking styles like political, news, or clerical speech are clearly determined by specific rhythms.

3.1.3 I-vectors

Identity vectors (i-vectors) have been introduced for speaker recognition and constitute speaker specific feature vectors. The idea is derived from *joint factor analysis (JFA)*, presented for instance by Kenny (2005), where a speaker *supervector* is decomposed in a speaker independent, a speaker dependent, channel dependent, and residual components. Speaker supervectors are concatenated *Gaussian Mixture Model (GMM)* mean values trained on acoustic feature vectors, typically on MFCCs. The UBM is a speaker independent global GMM.

If s is the speaker supervector, m is the speaker independent supervector (from the *Universal Background Model (UBM)*), V is the eigenvoice matrix, y are the speaker factors, U is the eigenchannel matrix, x are the channel factors, D is the residual matrix, and z the speaker-specific residual factors, then:

$$s = m + Vy + Ux + Dz \tag{3.3}$$

The individual components are focused on the respective principal dimensions, though the training procedure is extensive and will not be discussed here. More details can be found in Kenny (2005). I-vectors derive from the factor analysis, however, in comparison to the JFA approach, the speaker and the channel variability are combined, since it has been reported by Dehak (2009); Dehak et al. (2011) that the channel component of the JFA still contains speaker specific information. Therefore, as suggested by Dehak et al. (2011), the decomposition is rewritten as follows:

$$s \approx m + Tw \tag{3.4}$$

where s is the supervector, m is the speaker independent supervector (from the *Universal Background Model (UBM)*, as in Reynolds et al. (2000)), T is the total-variability matrix, as in Kenny et al. (2005), and w is the i-vector. The difference in the training of the V matrix in JFA and the T matrix from the i-vector approach is that the T matrix is trained treating all supervectors s as belonging to different speakers, as in Dehak et al. (2011). This means that all utterances are considered to belong to different speakers, and in this concrete case to different expressions. The similarity of concrete speakers/expressions can be derived calculating the distance between the vectors. This allows us to avoid labeling since we are interested in acoustic similarity of data, not in concrete predefined labels. Also, with respect to expressiveness, as repeatedly mentioned throughout this thesis, it is very difficult to define consistent and reliable labels for databases, where the expressiveness is not explicitly controlled, like for instance in audiobooks or interviews opposed to laboratory recorded corpora. In this sense, each supervector, i.e. each utterance, should be homogeneous with respect to speaker, channel, expressiveness, etc., i.e. belong to only *one* speaker, channel, expression, etc.

Usually, in speaker verification, i-vectors are compared calculating the cosine distance. If they point in the same direction, i.e. $cos\_dist = 1$ they are considered to perfectly belong to the same speaker. A threshold value would be used to determine the minimum possible cosine distance for the final decision.

In expressive speech, the usage of i-vectors is recent. I-vectors have been used for the first time by Lopez-Otero et al. (2014) in emotion recognition, where they represented emotional input waveforms which were classified on a continuous scale, as suggested by Kehrein (2002) (refer to Section 2.5 for more details), for evaluation. Then, they have been used for unsupervised data clustering of a multi-speaker database (audiobook) by Jauk et al. (2015). The main motivation to use them in expressive speech is to use them especially in multi-speaker databases, where not only the expressiveness has to be taken into account, but also the different speakers. The resulting data clusters have been used for expressive voice training, as presented in Section 3.2.2. The audiobook is not a multi-speaker database per se, since the book characters are acted by the same speaker. However, it can be considered as an approximation to a multi-speaker environment, where i-vectors outperformed all other feature combinations. Jauk and Bonafonte (2016b) expanded this paradigm in a study with different feature combinations, building i-vectors on pitch, intensity and syllable duration, as exposed in Section 3.2.3. Also, multi-speaker and mono-speaker databases have been compared yielding results which support the usefulness of i-vectors in multi-speaker environments. Finally, the objective results of these experiments are compared to state-of-the-art feature sets extracted with OpenSMILE in Section 3.2.4 in order to prove their suitability for the representation of expressive speech in multi-speaker (or approximated multi-speaker) environments.

3.1.4 OpenSMILE

The *Munich open-Source Media Interpretation by Large feature-space Extraction (OpenSMILE)*, by Eyben et al. (2013), is a feature extraction toolkit used for different tasks in speech representation, often used to extract features for emotional/expressive speech. The interesting aspect of this feature extractor is the exhaustive number of different features and statistics about them which can be extracted. Feature sets extracted with this toolkit have been used in many expressive and emotional speech related tasks, like emotional speech challenges (see Section 3.2.4 for more details). Some feature sets are considered to be state-of-the-art feature sets for emotional speech and therefore are a good comparison base for i-vectors and other features sets proposed in this work. A list of features which can be computed by openSMILE is provided in Table 3.1, as in Eyben (2016).

In the experiment described in Section 3.2.4, certain specific feature combinations are used for comparison, which will be referenced respectively. These feature combinations have been used in emotion recognition challenges and similar tasks, being state-of-the-art features for the representation of emotional speech.

Table 3.1: Low-level audio features by OpenSMILE.

Frame energy
Frame intensity/loudness (approximation)
Critical band spectra (Mel/Bark/Octave, triangular masking filters)
MFCC
Auditory spectra
Loudness approximated from auditory spectra
Perceptual linear predictive coefficients and cepstral coefficients
LPC
Line Spectral Pairs
Fundamental frequency
Voicing probability
Voice-quality: Jitter and Shimmer
Formant frequencies and bandwidths
Zero- and mean-crossing rate
Other spectral features (arbitrary band energies, roll-off points, centroid, entropy, maxpos, minpos, variance, skewness, kurtosis, slope)
Psychoacoustic sharpness, spectral harmonicity
Octave warped semitone spectra (CHROMA) and energy normalized and smoothed CHROMA (CENS)
CHROMA derived features for chord and key recognition
F0 harmonics ratio
Statistical and other functional features such as means, variances, extremes, ranges, regression, durations, peaks, onset/offsets, etc.
Classifiers like voice activity detection (VAD), GMM and NN based classifiers, pre-trained emotion recognition models (openEAR), etc.

3.2 Experiments

The role of acoustic features in emotion recognition or similar tasks is straightforward, but how can they be useful in speech synthesis? Large (multi-speaker) databases, like audiobooks, TV shows, interviews, radio, etc., are a rich source of expressive speech. However, the larger the database is, the more difficult is to label it in order to identify needed data. Often, it is even not very clear, how exactly to label such "uncontrolled", "real-life" data, which often contains speech of multiple speakers, many nuances of many expressions, speaking styles, etc. The solution to this problem is apparently automatic processing of data. The idea is to automatically cluster data into homogeneous speaking styles/expressions/emotions, avoiding labeling at all. In order to do it in the acoustic domain, robust features are needed which actually effectively represent expressiveness. This section aims at comparing the suitability of different feature sets for this task. Among others, the proposed features for the multi-speaker domain are the i-vectors, which will be compared to more traditional features. The databases, the experiments are carried out with, are actually not really "real-life" data, but an audiobook, rich in expressive speech and imitations of different speakers; and a studio-recorded emotion corpus spoken by two actors, a male and a female one. This last corpus is used for comparison, in or-

der to identify how different feature sets work for different data, mono-speaker, and the audiobook is approximated multi-speaker corpus (acted characters).

3.2.1 Experimental framework

The basic idea of the experiments is, calculate features from audio and use them to make data clusters. Of course, a classification could be carried out, where some classifier mechanism would learn how to predict expression classes from feature sets. However, as argued in different occasions in the thesis, it is very difficult to define a reliable label set for a database where the number of expressions, speaking styles, etc. is very exhaustive. Also, the manual effort to label such a corpus would be enormous. On the other hand, to train a statistical classifier, enough data is needed for each expressive class, which is again difficult for such a database. So, the goal is to provide an unsupervised method, where no labels are needed to create data clusters of expressively homogeneous speech.

Before continuing with the experimental framework, a corpus description will be provided in order to facilitate the understanding of the framework and the evaluation process. The first corpus is a juvenile narrative audiobook recorded in European Spanish, with a total of 7900 sentences, and of 8.8 hours of duration. It is spoken by one speaker who acts and imitated different characters, expressions, etc. Bad utterances are identified partly by automatic tools and partly by manual revision and removed. The second corpus is a laboratory recorded emotional speech corpus in European Spanish, recorded by two professional actors, male and female, with approx. 350 sentences each, recorded 7 times by each speaker, each time with a different emotion, a total of 6.4 hours of duration, as by Hozjan et al. (2002). The emotions recorded in the database are *angry*, *disgust*, *fear*, *joy*, *neutral*, *sadness* and *surprise*. Both corpora are used differently in the experiments and the details will be given for each experiment in the respective sections.

What is done in the first step is, for a given corpus, for each utterance a feature set is calculated, which is then fed into a clustering algorithm for unsupervised clustering, in this case k-means, concretely *VQ*, as by Gray (1984). The implementation used in this work are the VQ/LBG tools from the *SPTK toolkit*. Once the clusters are formed, they are evaluated. The evaluation happens by measuring the cluster entropy. If a cluster is expressively homogeneous, the information content, i.e. the entropy would be low. On contrary, if a cluster has many different expressions, it would contain more information and the entropy value would be high.

How is the information level measured if there are no labels for the data in the cluster? For this purpose, and only for the evaluation, a small part of the audiobook has been labeled manually with expression labels, and with character labels. The labeling of expressions was not aimed at classification, it was carried out as detailed as possible, trying to catch nuances like expression combinations, intensity where it was especially notable, etc. Some examples of the labels are *surprise-anger* vs *surprise-joy*, *excited-happy*, *cold-angry*, etc. The character labels identified the speaking character, or the narrator, where applies. In all evaluations, the expressive labels and the character labels are treated independently, i.e. no cases are treated where for instance a particular character uses

different expressions. The reason for this exclusion is that, basically, there is not enough labeled data in order to reasonably cover particular expressions for a particular character, i.e. there only few cases where each character is for instance *sad*, *happy*, *angry*, etc. In this sense, since character labels are much easier to obtain than the expressive labels, they have been assigned to use them as an additional test criterion to see how the clustering behaves with different features if looking at the characters. In total, a set of 248 different labels of expressiveness and 18 characters was obtained. The labeled part contains 1200 sentences, of a total duration of approx. 1.5 hours. Since approximately half of the sentences was labeled as *neutral*, these sentences were removed in order to provide a certain equilibration, reducing the total amount of used sentences to 600 of approximately 1 hour of duration. With respect to the mono-speaker corpus, since it was recorded in studio, labels are intrinsic.

The measured entropy, as proposed by Shannon (1948), and applied in clustering e.g. by Zhao and Karypis (2004), is defined as:

$$H(X_c) = -\sum_{i=1}^{q} \frac{n_c^i}{n_c} \log_2(\frac{n_c^i}{n_c}) \tag{3.5}$$

where X_c represents the whole set of speech segments in a cluster c, q is the number of labels, n_c^i is an element in cluster c labeled with i and n_c is the number of elements assigned to the c^{th} cluster. Then, weighted entropy is calculated as:

$$\bar{H}(X_c) = \frac{\sum_{c=1}^{C} n_c \cdot H(X_c)}{N} \tag{3.6}$$

where N is the number of all elements in the database, C is the number of clusters generated by VQ, $H(X_c)$ is the entropy of the c^{th} cluster and n_c is the number of elements assigned to the c^{th} cluster.

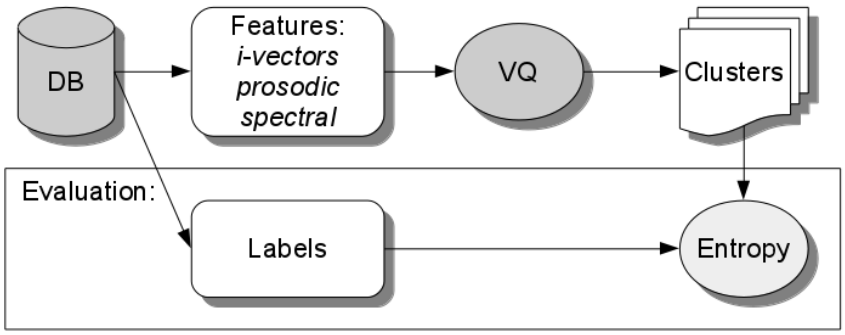
Additionally, to facilitate the reading, as an alternative measurement criterion perplexity was calculated. Perplexity is an expansion of the entropy frequently used in language modeling. It is defined as:

$$PP = 2^{\bar{H}(X)} \tag{3.7}$$

Figure 3.3 gives an overview of the clustering and evaluation process. First, from a database, a set of features is extracted (the features named in the image are rather symbolic, not always these types of features are extracted), then, a clustering algorithm is applied, VQ in this case, and a number of data clusters is formed. For the evaluation, for each of the utterances of the database, an expression, and where applicable (=audiobook), a character label is provided. The labels are then used to calculate the entropy of the clusters, indicating the "goodness" of each cluster.

Now, since the corpora do not change, nor does the clustering method, the variable parameter are the features. Better feature sets will yield better clustering results, showing higher suitability for given task. The evaluation method is further on referred to as objective evaluation. The objective evaluation is performed in three experiments, in chronological and evolutionary sense.

Figure 3.3: Clustering and evaluation framework.



1. I-vectors trained on MFCCs (=spectral i-vectors) are compared to other, "more traditional", features for the labeled part of the audiobook only.

2. Based on the satisfactory results from the first experiment, additionally to the spectral i-vectors, i-vectors are trained on prosodic and power features. Also, other features and feature combinations are compared between each other. This time, both databases are involved, the audiobook, acting as an approximated multi-speaker database, and the emotional mono-speaker corpus with both speakers.

3. Since i-vector-related features sets have performed very well for the multi-speaker corpus, an additional evaluation will be performed comparing the feature sets from the previous experiment to features sets extracted with OpenSMILE.

Additionally, in a side study, since different feature combinations are tested, weighted euclidean distance was compared to the normal euclidean distance in clustering. If euclidean distance is defined as:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.8}$$

where n is the number of dimensions of the space. Weighted euclidean distance can be defined as:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} w_i(x_i - y_i)^2}, \quad w_i \geq 0 \tag{3.9}$$

For this test, an algorithm systematically changed the weights for each feature in the feature vector for the distance calculation. Clustring was then performed

using the weighted distance, and the entropy was calculated for the clusters. However, the clustering for the best performing weight combination did not perform significantly better than the clustering based on the unweighted euclidean distance. For this reason, application of weighted euclidean distance was disregarded.

Now, retaking the motivation for the study of features for the representation of expressive speech, the results are applied to speech synthesis. Here the idea is, after having formed clusters of homogeneous speech data automatically, speech synthesis models can be trained using the data in these clusters.

Figure 3.4: Clustering and synthesis framework.

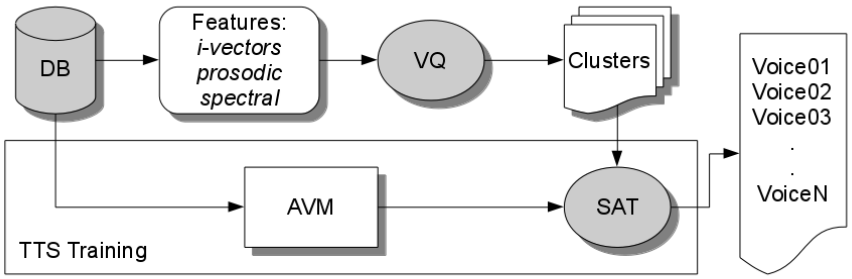


Figure 3.4 shows the framework for the synthesis from clusters. As for objective evaluation, a feature set is extracted for each of the utterances of the database. Here on contrary, only one feature set is extracted, the best one from the objective evaluation, i.e. the one with lowest entropy, to ensure the highest possible cluster homogeneity. The clustering process is applied to the features as for the objective evaluation. In parallel, an average voice model (AVM) is trained using the database. Then, using the data from each of the clusters and the AVM, speaker adaptation is performed, such that for each of the clusters, an individual synthetic voice is trained and can be used to synthesize speech. Speaker adaptation is used rather than normal HMM-training especially because of the small amount of training data. This framework is evaluated in two subjective listening tests as described below. The subjective tests formed part of the first two experiments, which were carried out chronologically. The details and differences will be explained in the corresponding sections. In the third experiment, no subjective test was conducted since the goal of the test was rather the objective comparison of the proposed feature sets to the state-of-the-art features sets extracted with openSMILE.

For the evaluation of the synthetic voices, a web based framework was designed, where the subjects were asked to edit a paragraph of an audiobook choosing the most appropriate synthetic voice for each of the sentences of the paragraph, paying attention to the expressiveness and the character who is at talk. The interface is shown in figure 3.5.

Figure 3.5: Subjective experiment web interface.

First of all, an introduction is shown, which explains the experiment and gives some background information on the paragraph in order to provide the users with some context of the situation handled in the paragraph. This is important especially if the participants do not know the book story. Below, the names of the participating book characters are given. In the first experiment, the participants could play the original voices clicking on the button with their respective names, in the second one, there were no original examples. Below, a sample sentence with a neutral content could be reproduced for each of the synthetic voices. In the main body of the interface, on the left side, the paragraph sentences are listed. On the right side the participant could choose one of the synthetic voices to read the corresponding sentence. In total, 10 voices were available. Below the paragraph (not seen on the image) there is a button which allows the reproduction of the complete paragraph with the chosen synthetic voices. Once the participant is satisfied with her/his choice, the choices can be saved and a comment can be left. The idea is that if any voices are particularly suitable for concrete characters in certain situations, then there would be a clear preference of choice for these voices among the participants. Further details will be given in the corresponding experiment sections.

3.2.2 Experiment 1: MFCC i-vectors and a small corpus

The corpus used in this experiment is the labeled excerpt of the audiobook described in the section above. It corresponds to the first four chapters of the book with a duration of approximately 1 hour. Neutral-labeled speech has been removed. A total of 247 expression labels (248 - *neutral*) and 18 character labels remained in the corpus. The corpus is segmented using the Ogmios speech synthesis tools, as by Bonafonte et al. (2006). A segment is defined as an orthographic sentence. In cases where the expressiveness varies throughout the sentence, or where the sentence is a combination of direct and indirect

speech, the segments are as long as the expressive style remains unchanged. 600 segments were obtained in total.

Features

The feature definition is based on prosodic and spectral criteria, that are both considered relevant in order to account for different speaking styles, characters and expressions. The prosodic criterion includes rhythmic features, accounting for rhythm driven speech styles such as news reading or rhythmically marked characters. Also MFCC based i-vectors are used, motivated by results achieved by Lopez-Otero et al. (2014).

Following features were taken into account.

- Prosodic features:
  - Intonation: means, range and Bezier polynomial coefficients, as proposed by Escudero et al. (2002), of the pitch were computed using Ogmios tools Bonafonte et al. (2006).
  - Rhythm: silence and syllable rates (#/sec), duration means and variation, computation based on segmentation.
- Spectral features:
  - Formants F1-F3 means for Spanish vowels /a/, /o/, /u/, /i/, /e/. These features, were extracted using Praat Boersma and Weenink (2015).
- I-vectors of dimension 600 were extracted using a UBM with 512 Gaussian components. The i-vector dimension might seem elevated, but on experimental basis we found that it performed best. As acoustic feature, 40 Mel-frequency cepstral coefficients were extracted using the AHOCoder, as proposed by Erro et al. (2011a). Before extracting the i-vectors, a *Universal Background Model (UBM)* and the total variability matrix are trained as described in Reynolds et al. (2000) and Kenny et al. (2005), respectively. Half of the labeled chapters of the audiobook corpus is used for each training. The i-vectors are calculated using *Kaldi* software, as by Povey et al. (2011).

  The training data was automatically divided into segments using a *voice activity detector (VAD)* disregarding silence intervals, such that all features were extracted disregarding silence (except for the silence frequency and duration features). When combining all features, 626 dimensional vectors were obtained.

Objective results

A total of 64 clusters were obtained with the k-means algorithm. Different experiments were carried out with different cluster numbers, this concrete number offered a good balance between data distribution throughout the clusters and data density for each cluster. Because when the number of clusters was too low,

they were too heterogeneous, when it was too high, the was too little data in each.

Table 3.2 gives an overview of the perplexities calculated for the clusters. The cluster perplexities are compared to the perplexity of the database, which is considered the worst case scenario.

Table 3.2: Perplexities (PP) for silence rate, syllable rate, mean F0, F1-F3 for /e/ , F3 for /o/ and i-vectors for Expressions (Ex) and Characters (Ch) in comparison to the database.

	PP/Ex	PP/Ch
DB	140.4	8.3
$silRate$	10.6	4.7
$sylRate$	9.4	4.0
$meanF0$	9.8	4.2
$F1-F3(e)$	10.1	4.0
$F3(o)$	7.1	3.7
$i-vectors$	9.0	3.5
all	10.5	3.6

As shown in the Table 3.2, the perplexity of the created clusters is significantly below the perplexity of the original database. The table also shows clear improvement when i-vectors are used for the discrimination of expressiveness and, especially, characters. An interesting behavior is observed when using the third formant of the vowel /o/, differently from other formants and other vowels. The expression perplexity lowers to 7.1 while character perplexity lowers to 3.7. However, the F3 of the other vowels do not show such low perplexity values. This effect might be speaker or database dependent. The results obtained with the individual formants of the other vowels are similar to the results of the three formants of /e/, which is shown in the table as an example.

Also important is the result achieved with the syllable rate. It is not better than the i-vector result, but better than the F0 result. Also, in further experiments in the multi-speaker domain, the good performance of this rhythmic feature will be confirmed, especially in combination with other features.

Details on the first subjective experiment

As introduced above, the cluster data can be used to train voice models. For this purpose, speaker adaptation is performed on the data from the 64 clusters which are formed with the best feature set from the objective evaluation, i.e. i-vectors, though in this case it might better be called “expression adaptation”. Speaker adaptation is preferred before the normal training not only because of the small corpus size, but also because in each cluster there is even less training data. The average voice model was trained using the neutral labeled speech data, which was removed from the clustering process, containing approximately half an hour of speech, here, the more appropriate name might be “neutral voice model”. The adaptation was performed then for each of the 64 clusters. The resulting synthetic voices were judged between three developers. They

Table 3.3: Paragraph sentences of the first subjective experiment.

(1)	La profesora McGonagall sacó un pañuelo con puntilla y se lo pasó por los ojos, por detrás de las gafas.
(2)	Dumbledore resopló mientras sacaba un reloj de oro del bolsillo y lo examinaba.
(3)	-Hagrid se retrasa.
(4)	Imagino que fue él quien le dijo que yo estaría aquí.
(5)	-Sí
(6)	-dijo la profesora McGonagall-.
(7)	Y yo me imagino que usted no me va a decir por qué, entre tantos lugares, tenía que venir precisamente aquí.
(8)	-He venido a entregar a Harry a su tía y su tío.
(9)	Son la única familia que le queda ahora.
(10)	-¿Quiere decir...?
(11)	¡No puede referirse a la gente que vive aquí!
(12)	-gritó la profesora, poniéndose de pie de un salto y señalando al número 4-.
(13)	Dumbledore... no puede.
(14)	Los he estado observando todo el día.
(15)	No podría encontrar a gente más distinta de nosotros.
(16)	Y ese hijo que tienen...
(17)	Lo vi dando patadas a su madre mientras subían por la escalera, pidiendo caramelos a gritos.
(18)	¡Harry Potter no puede vivir ahí!

chose manually four voices, among all clusters, as to fit best the conversational situation and the characters who appear in the test dialogue from their point of view. The other six voices were chosen randomly. The dialogue was presented in the web interface as described above and contained the 18 sentences/phrases listed in table 3.3. For each sentence one of the 10 synthetic voices could be chosen, and the resulting paragraph saved.

Subjective results

A total of 19 persons participated in the experiment, most of them experienced with speech technology. Table 3.4 gives an overview of the perceptive experimental results.

The results suggest clear preferences of voice choice for the three characters: Voices *v0* and *v7* for narrator $Narr$, voice *v1* for character $Ch2$ and voices *v4* and *v6* for character $Ch3$. Characters $Ch2$ and $Ch3$ have more distributed values; this might be due to an issue commented by many participants, namely that sometimes it was rather difficult to identify from the text which sentence was spoken by which character. One of the participants always preferred the neutral voices for the whole passage.

Table 3.4: Relative preferences for the voices v0-v9 over the whole paragraph for the two characters (Ch2 and Ch3) and the narrator (Narr).

	v0	v1	v2	v3	v4
Narr	**0.55**	0.00	0.03	0.00	0.01
Ch2	0.01	**0.60**	0.07	0.07	0.01
Ch3	0.03	0.10	0.05	0.20	**0.29**
	v5	v6	v7	v8	v9
Narr	0.01	0.01	**0.37**	0.00	0.00
Ch2	0.00	0.00	**0.11**	0.04	**0.12**
Ch3	0.08	**0.17**	0.04	0.03	0.02

3.2.3 Experiment 2: Prosodic i-vectors and single- vs multi-speaker

In the second experiment two databases are used. The first one is a laboratory recorded emotional speech corpus in European Spanish, as described above. In the results, the female corpus will be referred to as C_1, the male corpus will be referred to as C_2. The second database is exactly the same from the first experiment. The corpus will be referred to as A_l. All the other conditions are the same as for the first experiment.

Features

Motivated by the positive results using i-vectors from the past experiment, the feature sets used in this experiment are expanded. First, among the "traditional" features, Jitter and Shimmer, as well as power (intensity) are added to the feature sets. Regarding i-vectors, these are trained not only on MFCCs, but also on F0, syllable durations and power. The following features are used in this experiment:

- i-vectors calculated on:
  - 40 Mel-frequency cepstral coefficients, extracted using the AHOCoder Erro et al. (2011a). i-vector dimension: 600
  - Fundamental frequency of voiced segments, extracted using AHOCoder, where only the voiced parts are used for the i-vector extraction. i-vector dimension: 12
  - Power. i-vector dimension: 16
  - Syllable durations, calculated using forced alignment with *Ogmios* Bonafonte et al. (2006). i-vector dimension: 12
- F0 means, variance and range between the minimum and the maximum values. Extracted using AHOCoder.
- Syllable frequency and durations, means, variance and medians. Extracted from a forced alignment using Ogmios.

- Silence frequency and durations, means variance and medians. Extracted from a forced alignment using Ogmios.
- Local Jitter and Shimmer. Extracted using *Praat* Boersma and Weenink (2015).
- Power, mean and variance.

Also, different feature combinations are tested, some similar features combined and abbreviated as follows:

- *Pitch*: F0 means, variance and range.
- *Rhythm*: Silence and syllable frequency and durations, means, variances and range.
- *JShimm*: Local jitter and shimmer.
- *iVecC*: F0 and MFCC based i-vectors.

As before, for all acoustic features, including power and i-vectors, silences were removed with a VAD, measuring only speech. The different i-vector dimensions were chosen experimentally to be the best performing in each category.

Objective results

Table 3.5: Perplexities for different features combinations and for the three databases. The female part of the emotional studio corpus (C_1), the male part of the same corpus, (C_2), and the audiobook database (A_l) for expressions (E) and for characters (Ch) are shown.

	C_1	C_2	A_l(E)	A_l(Ch)
DB	7.0	7.0	140.4	8.3
F0 means, variance	3.0	**2.7**	9.6	3.8
Pitch	**2.9**	2.9	9.4	3.9
Power	5.0	5.1	11.5	4.8
Pitch - Power	4.9	5.1	13.4	4.9
JShimm	5.9	5.6	10.6	4.5
Rhythm	4.3	4.6	9.2	3.7
Rhythm - Pitch	3.2	3.0	8.7	3.4
Rhythm - Pitch - JShimm	3.2	3.1	8.6	3.4
MFCCiVec	6.4	6.3	9.0	3.5
F0iVec	4.3	4.2	11.2	4.2
PoweriVec	6.2	6.3	11.6	4.4
sylDuriVec	4.7	7.0	27.9	4.9
iVecC	6.2	6.0	8.8	3.8
Rhythm - iVecC	4.5	4.9	**8.2**	**3.3**
Rhythm - JShimm - iVecC	5.2	4.5	8.5	3.5

Table 3.5 shows the results for the objective cluster evaluation. There are many differences between the features, and between the corpora. The bold marked values are the best results obtained. For the laboratory recorded corpora the best results are obtained using just F0 or the Pitch combination. While the best results for the audiobook were obtained using the Rhythm and the i-vector combination. Rhythm seems to be more important for the audiobook than for the laboratory corpora. In fact, the Rhythm and Pitch and the Rhythm, Pitch and JShimm combinations achieve almost the same results as the Rhythm and i-vector combination. On the other hand, Rhythm alone performs worse than i-vectors alone, although better than Pitch and JShimm alone.

On the other side, i-vectors do not perform well applied to the laboratory corpora. It seems to be due to the fact, that the laboratory corpora have only one speaker each, while the audiobook has many speakers (although only approximated by imitation). The best results here were obtained using the Pitch parameters. For the i-vector part in the single speaker corpora, the best results were obtained with the F0 based i-vectors.

An interesting observation can be made examining closely the individual cluster results for the female laboratory speaker using the i-vectors based on syllable durations. Several clusters of approximate size of 30 to 40 utterances appear to be totally homogeneous, i.e. all labels in these clusters belong to the same emotion ($entropy = 0$), e.g. *angry*, *surprise*, *disgust*, etc. This distribution suggests that the female laboratory speaker uses rhythm as an important tool to communicate emotions. However, this not true for the male laboratory speaker nor for the audiobook reader. This results are supported by the clear perplexity values for the i-vectors based on syllable durations. Many other clusters of the female corpus, formed with the syllable duration i-vectors, are often formed of emotions which acoustically could belong together, such as *joy*, *angry* and *surprise*, or *sad* and *fear*. Although some emotions, specially *fear* and *surprise* often co-appeared with other emotions. This is not surprising since these emotions can easily combine with others, for instance one can be surprised positively, i.e. joyful, or negatively, with fear or anger. Also fear can be more aggressive, i.e. angry, or more neutral, or close to sadness.

Details on the second subjective experiment

The experimental design for the second subjective test is in generally the same as for the first one. The same interface is used, however, the original voice examples for the characters have been removed, such that the participants had no voice reference for the characters, making the task more difficult. Here, the choice is surely influenced by the fact whether a participant does or does not know the book, and also by her/his imagination of how a certain character should sound.

Also, by using a very small training corpus for the first experiment, the overall voice quality was reported to be relatively poor. Therefore, for the second experiment, the complete audiobook was used for training. The AVM was trained on the whole audiobook. The clusters were also formed from the data of the whole audiobook. So, for each of the 7900 audiobook utterances a feature vector was calculated, using the best performing feature set from the objective

Table 3.6: Paragraph sentences of the second subjective experiment.

(1)	**N:**	Entonces él lo sabía.
(2)	**N:**	La idea hizo que de pronto las piernas de Harry se tambalearan.
(3)	**V:**	-No seas tonto.
(4)	**N:**	se burló el rostro.
(5)	**V:**	-Mejor que salves tu propia vida y te unas a mí... o tendrás el mismo final que tus padres...
(6)	**V:**	-Murieron pidiéndome misericordia.
(7)	**P:**	-¡MENTIRA!
(8)	**N:**	gritó de pronto Harry.
(9)	**N:**	Quirrell andaba hacia atrás, para que Voldemort pudiera mirarlo.
(10)	**N:**	La cara maligna sonreía.
(11)	**V:**	-Qué conmovedor.
(12)	**N:**	dijo.
(13)	**V:**	-Siempre consideré la valentía...
(14)	**V:**	-Sí, muchacho, tus padres eran valientes...
(15)	**V:**	-Maté primero a tu padre y luchó con valor...
(16)	**V:**	-Pero tu madre no tenía que morir... ella trataba de protegerte...
(17)	**V:**	-Ahora, dame esa piedra, a menos que quieras que tu madre haya muerto en vano.
(18)	**P:**	-¡NUNCA!

evaluation, i.e. the combination of Rhythm and F0- and MFCC-based i-vectors, with a total of 620 dimensions. These features were used for clustering, also forming 64 clusters with the whole audiobook data.

Different from the first experiment, the most suitable four voices for the paragraph sentences were chosen automatically. Basically, the acoustic distance was measured from the acoustic feature vectors of the original sentences and the 64 cluster centroids. The closest clusters were chosen, and the voices built on the data from these cluster were used as considered to be the most suitable to synthesize the paragraph sentences. The other six voices were chosen randomly, as in the first experiment. The 18 sentences/phrases used in this experiment are shown in table 3.6. Also here, for each sentence one of the 10 synthetic voices could be chosen, and the resulting paragraph saved. Also, responding to the reported difficulty from the first experiment to identify, which sentence was spoken by which character, initial letters were provided in front of each sentence to facilitate the identification.

Subjective results

Table 5.11 shows the results for the perceptual experiment. Each number represents the percentage of how often a voice is chosen to represent a book character. Bold numbers indicate the highest preferences. A total of 11 subjects have participated in the experiment, 8 of them not familiar with speech technology.

Although there were no clues of how the original characters sound, certain voices are systematically preferred for certain characters, and also for different parts

Table 3.7: Relative preferences for the voices v0-v9 over the whole paragraph for the narrator (Narr) and the two present characters (Ch2 and Ch3).

	v0	v1	v2	v3	v4
$Narr$	**0.42**	0.06	0.00	0.03	0.04
$Ch2$	0.13	**0.16**	0.14	**0.23**	0.03
$Ch3$	**0.18**	0.13	0.13	**0.31**	0.00
	v5	v6	v7	v8	v9
$Narr$	**0.23**	0.04	0.10	0.06	0.01
$Ch2$	0.09	0.05	0.03	0.10	0.03
$Ch3$	**0.18**	0.00	0.00	0.00	0.05

of the dialogue. So for instance, the narrator voice is chosen differently for the beginning of the dialogue and for the middle part, where tension rises. The characters are being interpreted more freely, especially the second one. Although 6 of 10 synthetic voices had been chosen randomly, it does not mean that some of them can not represent the characters adequately, such that the participants could decide to use them for the composition of their paragraphs. In general, the first four ($v0$-$v3$) and the 6th voice ($v5$) were mostly preferred for the interpretation. The voices $v0$-$v3$ are the ones which have been chose by the distance calculation between the original sentences and the cluster centroids. None of the participants selected the neutral voice for all sentences ($v4$), although it had higher segmental quality.

The results show that, first, different voices are obtained from different data clusters, and second, the voices are suitable for different characters, and different situations. Surely, some clusters will yield voices which might be considered not suitable for nothing, however, in general, the results show that the proposed method works.

3.2.4 Experiment 3: Comparison to OpenSMILE

This experiment seeks to validate the usage of i-vectors, especially for multi-speaker databases, by comparing them with state-of-the-art feature sets extracted with openSMILE and used in tasks like emotion recognition challenges.

The databases used for this experiment are the same as those used in experiment 2, as well as the objective test criteria and the proposed feature combinations. The comparison is done with feature sets used in the *Interspeech 2009 Emotion Challenge*, *Interspeech 2010 Paralinguistic Challenge*, the *openSMILE emobase2010 reference set* (which is an improved version of the old emobase baseline and is based on the Interspeech 2010 set), and the *Large openSMILE emotion feature set*. To resume the experimental conditions, the databases, i.e. the labeled part of an audiobook and the two emotional mono-speaker databases, are segmented on sentence level, and for the utterances of each sentence a feature vector is extracted. Then, a k-means clustering is applied on the feature space forming homogeneous clusters, a total of 64. The clusters are evaluated calculating their entropy. Better clusters have low entropy. More suitable feature sets will yield better clusters. The following sections describe the openSMILE

feature sets and the experimental results.

OpenSMILE extracted feature sets

The following features sets have been extracted using the openSMILE extractor. For further details, such as number of coefficients, please refer to the *openSMILE Book*, by Eyben (2016), and to the reference articles cited for each feature set, as well as to the configuration files provided in the openSMILE toolkit.

- **The Interspeech 2009 Emotion Challenge feature set (*is09*)**: This set contains 16 low-level descriptors and statistics applied on them, resulting in a total of 384 features. The low-level descriptors are: *Root-mean-square signal frame energy, MFCCs, zero-crossing rate, voicing probability, F0.* The functionals are: *mean, standard deviation, maximum and minimum contour values, range, absolute positions of the maximum and the minimum values, slope and offset of a linear approximation of the contour, quadratic error of the difference between the linear approximation and the actual contour, skewness, kurtosis.* A more detailed description of the feature set and the challenge can be found in Schuller et al. (2009).
- **The Interspeech 2010 Paralinguistic Challenge feature set (*is10*)**: This set contains 34 low-level descriptors with 34 corresponding delta coefficients, and 21 statistical functionals applied to them. 19 additional statistical functionals are applied at the four pitch based low-level descriptors and their delta coefficients. In total, the set contains 1582 features. The low-level descriptors are: *Loudness, MFCCs, logarithmic power of Mel-frequency bands, line spectral pair frequencies computed from LPC coefficients, smoothed F0 envelope, voicing probability.* The functionals are: *mean, standard deviation, absolute positions of the maximum and minimum values, slope and offset of a linear approximation contour, quadratic error of the difference between the linear approximation and the actual contour, skewness, kurtosis, the first, second and third quartiles, interquartile ranges, outlier-robust minimum and maximum values of the contour, outlier-robust signal range, the percentage of time the signal is above (*$75\% * range + min$*) and (*$90\% * range + min$*).* The additional pitch related features are: *smoothed F0 contour, local Jitter, differential frame-to-frame Jitter (the Jitter of the Jitter), local shimmer.* The functionals mentioned above are all applied to the additional pitch related features except for the outlier-robust minimum values and the range. More detailed information on the feature set and the challenge can be found in Schuller et al. (2010).
- **The openSMILE emobase2010 reference feature set (*emobase*)**: This set is a further development of the "emobase reference set" and is based on the Interspeech 2010 set, including changes and improvements on some features. It also contain 1582 features and is the recommended reference set. More details can be found in Eyben (2016) and in the corresponding configuration files of the tool kit.
- **The large openSMILE emotion feature set (*emolarge*)**: This exhaustive set includes a large number of low-level descriptors, the corresponding deltas, and functionals, a total of 6552. More details can be

found in Eyben (2016) and in the corresponding configuration files of the tool kit.

The feature sets described here comprise what is being considered to be state-of-the-art feature sets in emotion recognition and related tasks. The *emolarge* feature set contains basically all available descriptors and is interesting to compare to smaller, but more carefully designed sets.

Experimental results

Table 3.8: Perplexities for different features combinations, including openSMILE, and for the three databases. The female part of the emotional studio corpus (C_1), the male part of the same corpus, (C_2), and the audiobook database (A_l) for expressions (E) and for characters (Ch) are shown.

	C_1	C_2	A_l(E)	A_l(Ch)
is09	3.8	3.9	10.5	3.9
is10	3.2	3.3	10.8	4.0
emobase	3.2	3.2	10.5	4.0
emolarge	4.5	4.7	8.8	3.7
DB	7.0	7.0	140.4	8.3
F0 means, variance	3.0	**2.7**	9.6	3.8
Pitch	**2.9**	2.9	9.4	3.9
Power	5.0	5.1	11.5	4.8
Pitch - Power	4.9	5.1	13.4	4.9
JShimm	5.9	5.6	10.6	4.5
Rhythm	4.3	4.6	9.2	3.7
Rhythm - Pitch	3.2	3.0	8.7	3.4
Rhythm - Pitch - JShimm	3.2	3.1	8.6	3.4
MFCCiVec	6.4	6.3	9.0	3.5
F0iVec	4.3	4.2	11.2	4.2
PoweriVec	6.2	6.3	11.6	4.4
sylDuriVec	4.7	7.0	27.9	4.9
iVecC	6.2	6.0	8.8	3.8
Rhythm - iVecC	4.5	4.9	**8.2**	**3.3**
Rhythm - JShimm - iVecC	5.2	4.5	8.5	3.5

Table 3.8 shows the perplexity results for the clusters created with the different feature sets, those proposed in experiment 2, and those extracted with openSMILE, including *is09, is10, emobase* and *emolarge*. As a reminder, smaller perplexity means better separation of speakers and characters, and therefore more homogeneous (=better) clusters.

For the mono-speaker databases: The openSMILE extracted feature sets *is09, is10* and *emobase* performed better than the i-vector based feature sets (lower part of the table). The *emolarge* set had a comparable performance. In comparison to other, "more traditional" features and feature combinations, the same three openSMILE sets performed better than the sets *Power, Pitch-Power, JShimm* and *Rhythm*, however, they could not outperform the combinations

Rhythm-Pitch, Rhythm-Pitch-JShimm, and especially *F0 means and variance* and *Pitch (F0 means, variance and range).*

For the audiobook, expressions: The openSMILE sets stayed approximately in the middle range in comparison to other features, here, in contrast to the mono-speaker databases, the *emolarge* set achieving the best results. In comparison to the traditional features, the emolarge set came close to the best performing combinations *Rhythm-Pitch* and *Rhythm-Pitch-JShimm*. In comparison to the i-vector based sets, the openSMILE sets could not outperform the best performing sets.

For the audiobook, characters: The panorama is similar as for the expressions, being *emolarge* the best set among the openSMILE combinations, performing similarly to the rhythm combinations among the traditional sets, and not achieving the performance of the best i-vector based sets.

3.3 Discussion

The goal of of this chapter was to study how suitable are different acoustic features to represent expressive speech and how they can be used for synthesis. Using different features combinations, unsupervised clustering is performed on expressive speech corpora, and the data from resulting clusters is used to train synthesis models with speaker adaptation techniques, allowing to substantially decrease the necessary amount of training data.

For the evaluation of the proposed methodology, a small excerpt of an audiobook was labeled, clusters were formed from the labeled audiobook data, and the labels were used to evaluate the clusters calculating the entropy. Lower entropy means homogeneous clusters suitable for model training. With this paradigm, different features and feature combinations, including i-vectors and openSMILE-extracted features sets, were compared, also using different databases.

Objective results allow different conclusions regarding the suitability of different features. First, for the speech corpora recorded in a controlled laboratory environment, it seems that the more traditional features, especially the pitch related ones, worked best. Surprisingly, the simple pitch combinations even outperformed the sophisticated feature sets extracted with openSMILE and used for emotion challenges and similar tasks. This is not true though for the audiobook, which also was recorded in studio environment, however, the reader interpreted not only emotions and expressions, but also characters, which converts the audiobook to an approximation to a multi-speaker database. In this case, i-vectors seem to be more effective, probably as a consequence of the presence of different imitated speakers (book characters). Here, no feature set could outperform the best i-vector based combinations. Probably, in a real multi-speaker database, where speakers are not imitated, i-vectors could be even more effective. For single speaker domain though, possibly, for each individual case the best feature combination needs to be found by labeling a small portion and testing. Of course, it has to be taken into account that the present results have been obtained in an unsupervised clustering framework which relies in simple euclidean distance calculations. Trained models probably would yield a different panorama.

Using the data clusters from the audiobook formed with the best feature combination, subjective experiments were carried out. In the experiments, participants could edit small paragraphs of an audiobook using synthetic voices trained on the data clusters. For this, for each paragraph sentence 10 synthetic voices were made available to choose from and to design the dialogues taking into account characters (=speakers) and expressiveness.

The experimental results show that the proposed method permits to define expressive voices which can be chosen manually to synthesize expressive text in applications like audiobook editing. Although being practically useful, this method does not permit though automatic expressive voice assignment, since it is based on acoustic features only and there is no connection to the expressiveness in text, except by human intervention. The next chapter analyses methods of semantic vector representation of text which can be used to predict acoustics, providing a completely automatic method for expressive synthesis directly from plain text.

Chapter 4

Semantics-to-Acoustics Mapping

In the previous chapter, unsupervised clustering was performed in the acoustic domain on different feature sets extracted from corpora in order to gain expressively homogeneous training data automatically. This chapter deals with a similar task in the textual or semantic domain. The advantage of using text instead of acoustics to find expressive data is that text is much more easily available and processable than acoustics. Furthermore, the task by itself is not bare clustering of training data. The goal is to predict expressiveness from text. This can yield interesting applications such as automatic reading of expressive text like books or finding expressive or emotional data in large databases without acoustic feature extraction. In order to do that, numerical representations of text in functions of semantics, or even sentiment, will be derived and used to predict acoustic information, or directly as additional input features to a TTS system (please refer to Chapter 5 for further details on this task). This chapter describes these semantic representations, how they are related to expressiveness, and how acoustic information can be derived from them.

Before continuing with semantic representations, the question *What is semantics?* needs to be discussed. As Klabunde et al. (2004) states, *semantics* is part of *linguistics* and treats the meaning of textual units, as words, phrases, whole texts etc. In semantics, the meaning of these units is usually identified as *true* or *false*. The meaning of complex units is derived from logical relations between smaller units. Clearly, the *truth*-value is not the only information codified in text. There is also information derived from the context, world knowledge, information about the speakers, the relationships between speakers etc. All that type of information, not only semantic, allows humans to read texts, such as books, in an expressive and emotional way. Certainly, the manner of interpretation in reading might be very personal, however, we as listeners can very well intuit if the emotion transmitted by the reader is in accordance with the semantic meaning of the read text. The same could be claimed for spontaneous speech. Where this relation fails, i.e. the perceived semantic and emotional information is not in accordance, the emotional meaning might change creating expressive states like *irony*. According to this reasoning, it can be assumed that,

at least to some extent, some information about expressiveness of the text can be deduced from semantics.

The goal is to find out how expressive information can be derived from semantics and how it can be traduced to acoustics. There are *key words* or *clues* that indicate how an utterance might be pronounced. For example, the sentence *Happy birthday!* might be pronounced with a happy, or at least positive sounding voice, while *My grandmother has died.* probably will be pronounced with a sad voice. Of course there are cases where this type of clues does not work. These cases might depend on context, being associated with irony or sarcasm, or basically being exceptions with no clear explanation. In this work, the expressiveness which is intrinsically codified in the semantic of clues is referred to as *default expressiveness*, and in the case of exceptions it is referred to as *pragmatic expressiveness*. Given a text, where some words are clues where expressiveness can be derived from, what about all the other words which do not contain information about expressiveness? The idea is, which will be pursued on the course of the here proposed approach, is that even if a word or a word combination does not contain information about expressiveness, but does **co-occur** with a word or word combination which does contain it, in many cases the semantically non-expressive words will be articulated with a similar expressiveness as the expressive ones. The key word here is **context**.

It is relatively easy to identify expressive speech using key words, assumed that it is pronounced the default way. However, it is much more difficult to determine the pragmatic expressiveness of a text since it requires some codification of the relations between different contexts, or persons, or world knowledge. It might be for instance, that the sentence *The house is green.* is pronounced with disgust, because the person who says it finds green color disgusting. However, the sentence by itself does not indicate that type of information. So in order to predict the correct expressiveness for this sentence the system needed to know that the speaker hates green. That fact can possibly be deduced from context, or also be part of the world knowledge. How far that knowledge can be made available to the machine is a very difficult issue.

Several problems arise with working with structures like keywords. First, their potential number is infinite. Second, they need to be identified reliably in the text and interpreted by the system correctly in order to deduce information about expressiveness. This might be realizable for a limited domain, but impossible for open domains such as books. Rather, an efficient and numerical representation of text is needed such that terms can be organized automatically in a meaningful semantic way. Then, assuming that acoustic features related to expressiveness can be deduced from the semantic representation, regression models can be trained in order to predict acoustics from semantics.

A useful semantic representation is the *vector representation* or *term embeddings*. Here term, i.e. letters, words, n-grams, phrases, sentences, etc. are codified as multidimensional vectors. The relative position of the vectors reflects the semantic relations between the terms they represent. So, if two terms have exactly the same semantic meaning, they would be represented by the same vector, or by two vectors which are located very close to each other.

Semantic vector representations have been used in speech synthesis mainly in two manners. First, as an additional linguistic feature, especially in neural net-

work based speech synthesis (see Section 2.4.2 and Chapter 5). Second, to cluster or classify training corpora. For instance, Watts (2012); Alías Pujol (2006); Lorenzo Trueba (2016) used semantic vector representations for unsupervised clustering of training or adaptation corpora for speech synthesis. Eyben et al. (2012) also use semantic vectors to cluster training data and to classify input text for a TTS.

This chapter is organized as follows: Section 4.1 introduces semantic representations focusing on numerical representations. First, bag-of-words models are introduced and shortly discussed. Also, distance measures are discussed since these are used for basic classifications in vector space where the distance between the data points determines their similarity. Then, in Section 4.1.3, latent semantic indexing (LSI) is introduced and exaplained and visualized on a toy example. Further, a preliminary analysis of a labeled portion of an audiobook is conducted applying LSI and visualizing data distribution in term of expressive labels, in Section 4.1.3. Section 4.1.4 introduces the Skip-gram embeddings derived from neural networks, while Section 4.1.4 introduces embeddings calculated from the sentiment (positiveness/negativeness) of a sentence, also using neural networks. Section 4.2 proposes a method of how to use semantic embeddings to predict expressiveness (acoustics), and Section 4.3 describes the experiments based on the proposed approach with the corresponding results. Finally, in Section 4.4, the studied approaches and the results are discussed.

4.1 Semantic representation

Semantic analysis of text is a difficult task which, in the past, required a lot of manual effort. Words and concepts needed to be decoded semantically, in form of logical formulas or ontology-like models, where at least key-words had to be identified and made interpretable. A famous example for such a network of words is *WordNet*, by Princeton (2010). A famous tool to create own ontologies is *Protégé*, by Musen (2015). Such tools or networks provide an advanced semantic dictionary. To create a dictionary expert knowledge is needed, and this means, a lot of manual work.

How could semantics be analyzed automatically? As a first step, it needs to be codified numerically. Numerical codifying of text comes from *information retrieval*, where large amounts of text need to be processed automatically. The first idea is a basic *bag-of-words* representation, where a text corpus is understood as a set of documents $D = \{d_1, d_2, ..., d_K\}$, and each document is characterized by a set of words $W = \{w_1, w_2, ..., w_N\}$, which the respective document contains. A document can be of arbitrary size, i.e. can be for instance a sentence, a paragraph, etc. Words are also called *units* or *terms*. On the other side, a *unit* or a *term* can be of larger size than a word, for instance a bigram or a trigram. In this way, each document is represented by a vector of term occurrences and the similarity between terms and documents is derived from the proximity, i.e. it is assumed that similar documents will contain similar terms. A straightforward way to calculate the proximity is to use distance measures, some of them, which have been used in this work, are described in section 4.1.1.

These rather simple models are further developed into *Latent Semantic Indexing (LSI)*, which is based on *Singular Value Decomposition (SVD)* and is described

in section 4.1.3. Bellegarda (2012) proposes a similar method called *latent semantic mapping (LSM)* for unsupervised document clustering. LSI has been used for some preliminary experiments, codifying a labeled portion of an audiobook. The results are presented in the same section.

Most recently, in many language and speech processing areas, neural networks (see Chapter 2) are used for a variety of tasks. Also in numerical semantic representations, neural network based techniques have been developed. Sections 4.1.4 and 4.1.4 describe semantic *embeddings*, i.e. semantic vectors, which are derived from neural network layers. The neural networks are estimated with different training criteria, such that for example the *Skip-gram*, by Mikolov et al. (2013), model is trained to predict the probability of the context of a word, or the *Stanford sentiment model*, by Socher et al. (2013), predicts the probability of the positiveness or negativeness of a sentence. The vectors are then extracted from intermediate neural network layers.

4.1.1 Distance Measures

Generally, if a term is represented in a form of a vector, one of the most basic tools to calculate its similarity to other terms is to measure the distance to the other terms in the vector space. Usually for in semantic vector spaces, *cosine distance* is calculated as the dot product, as in:

$$d(X,Y) = cos(\Theta) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2} \cdot \sqrt{\sum_{i=1}^{n}(b_i)^2}} \tag{4.1}$$

where n is the number of dimensions of the space.

In this work, in a side study, different distance measures were compared, like the cosine distance, the Euclidean distance, the Canberra distance, by Lance and Williams (1967), and the Chebyshev distance, as by Deza and Deza (2009). The motivation for this comparison lied in the heterogeneous nature of the different features used for the experiments. The distance measures performed relatively similar, however, due to its simplicity and compatibility to unsupervised clustering operations, the Euclidean distance is used for the experiments presented in this chapter.

4.1.2 Bag-of-words Representations

The most simple technique of numerical representation of text is the *bag-of-words* representation. *Bag-of-words* is a representation of documents in terms of the words which constitute them. So, each document, which is a text of an arbitrary length and composition, contains words. Each word, also called *term* or *unit*, is counted. Each document is then represented as a vector of word (co-)occurrences, and the set of documents as $T \times D$ matrix, as in table 4.2, where T are the terms and D the documents.

An alternative way is the *inverse-index* representation. Here each word is indexed with the number of documents where it appears, although the background principle remains the same.

Table 4.1: Co-occurrence matrix. Columns are the documents, rows are the terms.

	d_1	d_2	d_3
t_1	1	1	0
t_2	1	1	0
t_3	2	0	1
t_4	1	0	5
t_5	1	0	0

In the basic version neither the order of words nor structure are considered. In some cases it can be a problem: for instance the utterances *The big dog is in the house.* or *The dog is in the big house.* are distinct semantically, however would be represented identically in a simple bag-of-words model.

Sparse Matrices and Term Weighting

Matrices are called *sparse matrices* because they usually contain a lot of zero values since many terms will not appear in all documents. On the other hand, there will be some terms which appear many times in all documents, like *stop words*, i.e. articles, conjunctions, etc. This makes the classification slower and possibly less reliable since many terms do not contribute useful information. There are several techniques that deal with this problem, as for instance by Klabunde et al. (2004).

- **Text preprocessing**
  In a first preprocessing step, all terms which are considered useless, like stop words, are removed. Additionally, words are normalized eliminating flections, often reducing words to stems.
- **Term selection**
  Only some terms which are considered to contribute the most information about the semantics are chosen.
- **Reducing the search space**
  There are different methods of search space reductions such as *clustering*, where the classification is achieved using cluster centroids. A different technique is the *Latent Semantic Analysis (LSA)*, that bases on *Singular Value Decomposition (SVD)*, see section 4.1.3. This technique reduces significantly the dimensionality of the search space.

After all eliminations and reductions, there still will be terms which are more important semantically than others. The most common way to deal with this issue is to weight the terms, according to Klabunde et al. (2004). A popular way of doing so is the *TD/IDF* technique. TF stays for *term frequency* and IDF for *inverse document frequency*. Basically, it is assumed that the importance of a term somehow is reflected in how often it appears in a document. Additionally, the relation between the number of documents that contain a specific term and

the total number of documents is taken into account. Let $t_{i,j}$ be the number of appearances of a term j in a document i, f_j be the number of documents that contain the term j and N the total number of documents, then the weight is calculated as:

$$w_{i,j} = t_{i,j} \cdot \log(N/f_j) \tag{4.2}$$

The interpretation of the weighting is, if a document occurs in all documents, the weight is equal 0, since it practically lacks semantic information, what happens for instance with articles, pronouns, etc. If a term occurs only in few documents, its importance depends on how often it occurs there, giving higher weights to higher occurrences.

Another weighting technique only considers the occurrence of a term without taking into account the frequency, the *boolean* technique. Other methods use the *entropy* of a term, others the *probability*, according to Klabunde et al. (2004).

When dealing with expressiveness, it is not fully clear if text preprocessing or term weighting contribute to classification or they do not. Studies conducted by Pennebaker (2011) report that especially stop words contain a lot of expressive information. In a preliminary study different vector space realizations have been compared and those that contained stop words achieved better results (see section 4.1.3). Word stemming, on the other hand, has always been applied in the experiments presented in this chapter.

Regarding term weighting, it is unclear how to apply it for expressive classification. Intuitively, term frequency is rather unimportant. Although there are terms which can easily be associated to certain expressions, there are a lot more which can be not. Key words could be weighted, though this again involves manual intervention.

4.1.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a technique of text representation, in terms of a vector representation, based on *Singular Value Decomposition*. If C is an $m \times n$ matrix, where $m > n$, then C can be decomposed as follows (as for example in Kuttler (2007)):

$$C = UDV^T \tag{4.3}$$

where U is the *eigenvector* matrix of CC^T, V is the *eigenvector* matrix of C^TC and V^T is its conjugate transpose, and D is the diagonal *singular value* matrix of CC^T and C^TC. Figure 4.1 illustrates the SVD with the matrix dimensions.

Applied to text, first, a co-occurrence matrix $C = m \times n$ is built, as in table 4.2, where rows are terms and columns are documents, using the following example corpus:

- *(d1)*: Harry Potter lives in Privet Drive number four.
- *(d2)*: Harry Potter has survived the attack of Voldemort.

Figure 4.1: Singular Value Decomposition illustration.

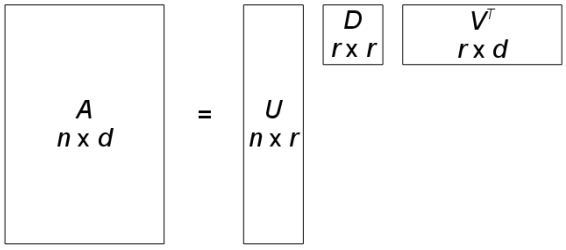


Table 4.2: Co-occurrence matrix

	d1	d2	d3
harry	1	1	0
potter	1	1	0
live	1	0	1
privet_drive	1	0	1
number_four	1	0	0
survive	0	1	0
attack	0	1	0
grandmother	0	0	1

- *(d3)*: My grandmother lives in Privet Drive.

Each sentence, i.e. document, is represented in a column, each term in a row. If a term occurs a times in a document, then $C[i, j] = a$. Now the singular value decomposition is applied as in equation 4.3, obtaining the matrices in equations 4.4, 4.5 and 4.6.

$$U = \begin{bmatrix} 0.476 & 0.310 & 0.144 \\ 0.476 & 0.310 & 0.144 \\ 0.433 & -0.427 & -0.077 \\ 0.433 & -0.427 & -0.077 \\ 0.293 & -0.091 & 0.530 \\ 0.183 & 0.402 & -0.386 \\ 0.183 & 0.402 & -0.386 \\ 0.140 & -0.335 & -0.607 \end{bmatrix} \tag{4.4}$$

$$V^T = \begin{bmatrix} 0.786 & 0.491 & 0.374 \\ -0.172 & 0.756 & -0.631 \\ 0.593 & -0.432 & -0.679 \end{bmatrix} \tag{4.5}$$

$$D = \begin{bmatrix} 2.684 & 0.000 & 0.000 \\ 0.000 & 1.883 & 0.000 \\ 0.000 & 0.000 & 1.120 \end{bmatrix} \tag{4.6}$$

The terms are represented in a new space by the row vectors of the matrix U and the documents by the column vectors of the matrix V^T. The singular values of the matrix D represent the dimensions of the vector space. The dimensionality can be reduced setting some singular values to zero. For visualization purposes the dimensionality is reduced to $k = 2$, obtaining:

$$U_2 = \begin{bmatrix} 0.476 & 0.310 \\ 0.476 & 0.310 \\ 0.433 & -0.427 \\ 0.433 & -0.427 \\ 0.293 & -0.091 \\ 0.183 & 0.402 \\ 0.183 & 0.402 \\ 0.140 & -0.335 \end{bmatrix} \tag{4.7}$$

$$V_2^T = \begin{bmatrix} 0.786 & 0.491 & 0.374 \\ -0.172 & 0.756 & -0.631 \end{bmatrix} \tag{4.8}$$

$$D_2 = \begin{bmatrix} 2.684 & 0.000 \\ 0.000 & 1.883 \end{bmatrix} \tag{4.9}$$

To obtain the vector space coordinates, the U_k and V_k^T matrices are multiplied by the singular values in D_k. For instance, to obtain the coordinates for the term *harry*, its vector in the U_2 matrix, which corresponds to the first row (since it is the first term), so $\langle 0.476, 0.310 \rangle$, is multiplied by the singular value diagonal matrix, as in:

$$\begin{bmatrix} 0.476 & 0.310 \end{bmatrix} \begin{bmatrix} 2.684 & 0.000 \\ 0.000 & 1.883 \end{bmatrix} = \begin{bmatrix} 1.278 & 0.584 \end{bmatrix} \tag{4.10}$$

Figure 4.2 visualizes terms (circles) and documents (squares) in the vector space. $w1$ corresponds to the first term (harry), with the coordinates calculated above, $w2$ to the second, which has the same coordinates since it has exactly the same vector in the co-occurrence matrix, etc. The coordinates for the documents are calculated the same way, but multiplying the columns of the V_2^T matrix with the singular values.

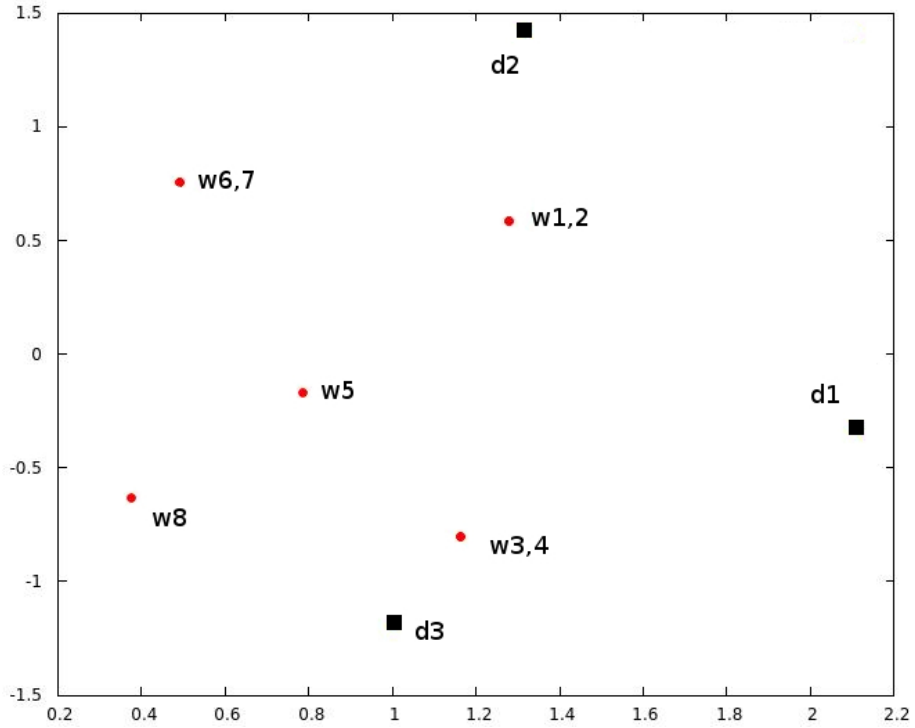
It can be observed that, for instance, the first two terms (harry, potter) are located between the documents $d1$ and $d2$, where they actually can be encountered in the corpus. The terms $w3$ and $w4$ (live, privet_drive) are located between the documents $d1$ and $d3$, as well corresponding to the corpus. Terms $w6$ and $w7$ (survive, attack) are closest to $d2$.

The next sections describe a preliminary experiment and results where a labeled part of an audiobook is projected into an LSI space and visualized. The labels represent expressiveness of the text.

Preliminary experiments with LSI

Given the assumption that some information about expressiveness is codified in text and can be deduced from semantics, semantic vector representations can

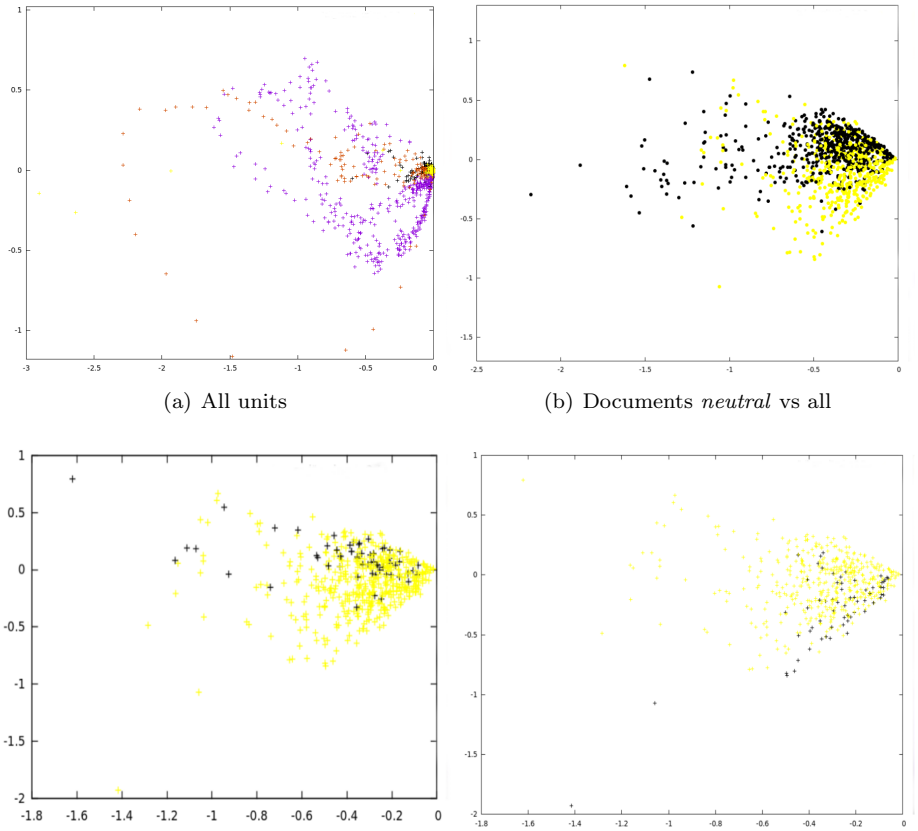
Figure 4.2: SVD Example



be a useful tool to automatically cluster data for unsupervised learning or for classifying input text. To analyze this, an SVD tool has been implemented using *GNU GSL*. The same labeled portion of the audiobook as in the experiments in Chapter 3 is used for this analysis. The first four chapters are labeled with expressive labels, where sentences have1 been split on direct and indirect speech breaks, each of them representing a document. The labels are designed to describe the expressiveness as well as possible with labels like *surprise-happy* or *excited-angry*, trying to describe all the nuances of expressiveness. This fine-grained labeling results in very unbalanced classes where the largest one, labeled as *neutral*, contains almost the half of the documents and the next lower classes contain only few dozens of documents, resulting also in many (almost) "hapax legomenon" classes. In order to make the statistics more robust many classes are eliminated summarizing them and reducing the total number to about 200. However, it must be noted that the summarization is a rather difficult task given the great variety of expressiveness and characters in the audiobook.

A total of 1 079 documents, containing 25 742 terms: words, bigrams and trigrams, are used to train the vector space. As done before in the toy example, the dimensions of the vector space are reduced to 2 in order to visualize it. Some example visualizations are shown in figure 4.3.

Plot 4.3*a* shows the distribution of all units, where the color indicates the label of the document the unit belongs to. The distribution is highly nonlinear, it



(a) All units

(b) Documents *neutral* vs all

(c) Documents *suspense* vs all (no *neutral*)

(d) Documents *exclamation* vs all (no *neutral*)

Figure 4.3: LSI Expression plots.

has a point of concentration, which contains the most general terms, and spiral-like arms. It can be assumed, the farer away a unit from the center, the more specific is its meaning. Plot 4.3*b* shows the distribution of documents with the label *neutral* and the rest of the labels. There is a relatively clear separation of these two types of documents. Plots 4.3*c* and *d* show the same type of separation for *suspense* and *exclamation* labeled documents, respectively. Here in *c* and *d*, *neutral* labeled documents have been excluded in order to balance the visualization, because almost 50 per cent of the documents are labeled as *neutral*. Spatial separations can also be observed here, though a little less clear than for the *neutral* labeled documents. In all plots there is a center of concentration, the rest of the documents are distributed in a kind of an arrow-like shape towards this center.

Since some separation is observable on the plots, a small preliminary experiment is conducted where a classification of documents (=sentences) is tried to be performed. For this, a random test set of 121 sentences is excluded from training of the vector space and is used for the classification task. The classification is performed calculating the distance of each test sentence to its N nearest neighbors (documents). The test sentence is assigned to the class K, to which

most of the nearest neighbors in a radius R belong to. The radius was set to 0.1, though different experiments with different radii were conducted, introducing variations especially to better account for documents which are farer away from the center and there is more space between the neighbors. This classification task is a preliminary experiment. In order to train a more sophisticated model more data is needed.

As to the results of the preliminary experiment, in the best cases where the test sentences are part of a "large" class, the correct classification is as high as 70 per cent, i.e. a 70 per cent of the samples belonging to a large class were classified correctly. Large classes are, for instance, classes labeled as *neutral* and *suspense*. However, the classification of "smaller" classes is not successful with values going down to only about 10 per cent of correct classification. Since the test set is random, the half of the test samples belongs to *neutral* labeled classes, and regarding the next bigger classes, about 70 percent of the data belongs to only 5 classes. When the neutral class is removed, the overall classification accuracy is not higher than 10 per cent, where a closer look is unnecessary.

It seems that semantic vectors do contain information which can be interpreted as expressive, though with limitations. On the one hand, only test sentences belonging to big classes could be assigned more or less reliably. On the other hand, probably, to achieve a good class separation much more data is needed to train the semantic model, and a more sophisticated classifier.

4.1.4 Continuous Semantic Embeddings with Neural Networks

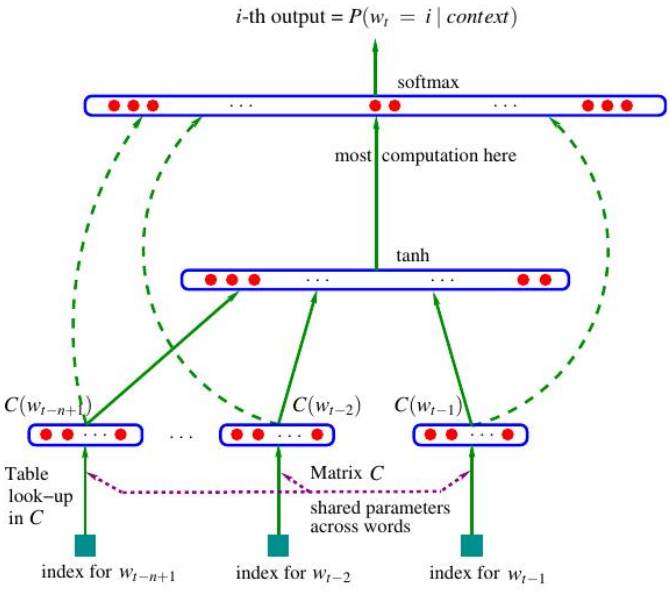
An alternative approach to LSI for semantic vector representations is to use neural networks. The basic idea is to train a neural network with a certain *training criterion* and to extract the term vector representation, called *embedding*, from the output or an intermediate layer, as for example proposed by Bengio et al. (2003). *Term* in this case is not limited to words or bigrams, trigrams, etc. The embedding can be trained also for arbitrary units such as letters, words, phrases, sentences, etc. In the following section it is explained how these representations can be improved. A popular model was proposed by Bengio et al. (2003) for learning jointly word vector representations and a statistical language model. The proposed network is a feed-forward architecture and consists of an input layer, a linear projection layer, a hyperbolic tangent hidden layer and an output *softmax* layer, where the training criterion is to predict word probability as in:

$$\hat{P}(w_i|w_{i-1},...w_{i-n+1}) = \frac{e^{y_{w_i}}}{\sum_{j=1}^{n} e^{y_{w_j}}} \tag{4.11}$$

where y_i are the unnormalized log-probabilities for each output word i. The input are N words with a *1-of-V* encoding where V is the vocabulary size. The projection matrix C is shared for all words. The input is the index of each word i. The architecture is illustrated in figure 4.4.

Mikolov et al. (2013) propose a derived architecture, but reducing significantly the computational complexity. Their proposed architecture is similar to the one by Bengio et al., but they remove the non-linear hidden layer and share the

Figure 4.4: Neural Network based language model, from Bengio et al. (2003).



projection layer among all words, not only the projection matrix. Concretely, they train two models, the *Continuous Bag-of-Words Model (CBOW)* and the *Continuous Skip-Gram Model*. The CBOW model predicts a current word w_t from the context of that word in a context window of length c. On contrary, the Skip-Gram model predicts the context words of a current word w_t maximizing the average log probability as in:

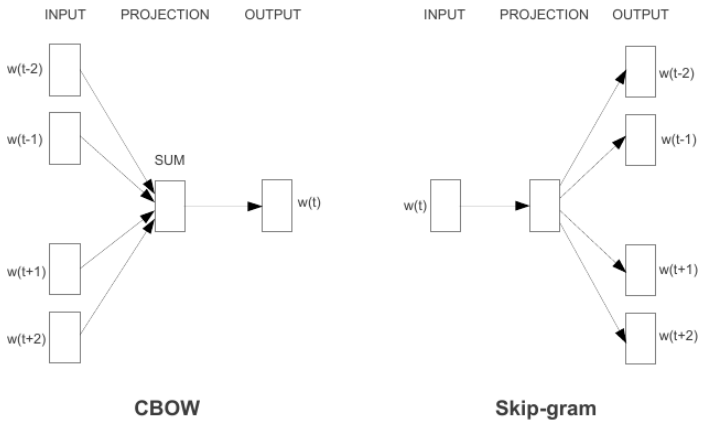
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{4.12}$$

The range of the context improves the quality of the resulting word vectors, but also increases the computational complexity. Also, words in the context window can be "skipped" when the prediction probabilities are too low. Figure 4.5 shows both system architectures in comparison. It has to be noted that the context words are predicted individually, not all at once, i.e. the output is always one context word. On contrary for CBOW, the input is constituted by all context words.

Mikolov et al. (2013) improve the Skip-Gram model by adding a method for creating embeddings for phrases such that for instance combinations like "New_York" are considered to be one term and not a combination of isolated terms "new" and "York".

An alternative method which achieves good results is *global vectors for word representation (GLOVE)* proposed by Pennington et al. (2014). This method uses word co-occurrence statistics to train the models and results in similar word

Figure 4.5: CBOW and Skip-Gram architectures, from Mikolov et al. (2013).



vectors as Skip-Gram.

Section 4.2 proposes a method to predict expressiveness from acoustics and section 4.3 presents three experiments and results based on that method. In all three experiments, the Skip-Gram model is used to train semantic word vectors and to create phrase and sentence embeddings, since it is considered to be better for rare words, which can be of advantage for expressive speech. In preliminary experiments, it was compared to the CBOW and achieved better results.

Standford Sentiment Analysis

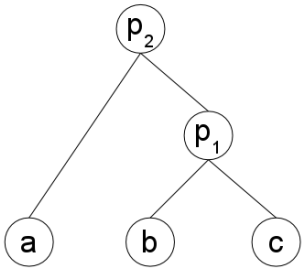
Socher et al. (2013) propose a recursive neural tensor network to create embeddings and to predict sentiment probabilities of terms. *Sentiment* is the valence, i.e. the positivity of the term. A term can be everything from word to sentence level. The network is trained on the labeled *Sentiment Treebank* which consists of a movie review database. The sentences are labeled as positive or negative reflecting the intention of the review publisher. Furthermore, reviews have been split in subphrases and annotated on a sentiment scale using Amazon Mechanical Turk. All sentences are parsed with the Stanford Parser, as by Klein and Manning (2003), and stored as binary trees. The word vectors are initialized randomly and an RNN computes the parent vectors in the following hierarchy:

$$p_1 = f(b, c), p_2 = f(a, p_1) \tag{4.13}$$

where a, b, c are word vectors, $f = \tanh$ is a standard element-wise nonlinearity and p_i is the parent vector i. The hierarchy is illustrated as an example binary tree in Figure 4.6.

The input to the sentiment parser is a sentence, the output can be the sentiment value (positive, negative, neutral), the probability and the vector embedding of the sentiment for each binary node of the tree structure, from the top node

Figure 4.6: Example binary tree of the sentiment parser.

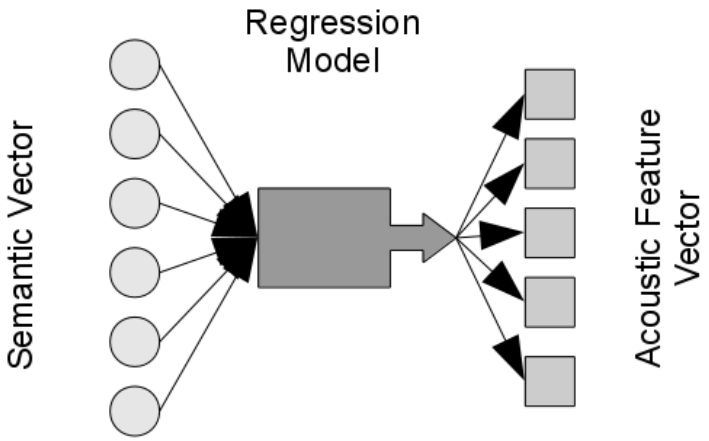


down to the word level. Chapter 5 presents a neural network based TTS which uses sentiment embeddings for expressive speech synthesis.

4.2 Predicting Acoustics from Semantics

Given a meaningful semantic representation of text, and the assumption that acoustics related to expressiveness can be derived from these semantic representations, a classification or regression model can be trained in order to predict acoustic classes or features. The here proposed method focuses on regression and vector-to-vector mapping of semantic vectors to acoustic feature vectors, as in figure 4.7.

Figure 4.7: Vector-to-vector mapping.



The mapping is a prediction task, where a statistical regression model learns to predict a vector from another. Many linear and nonlinear predictor models

exist, like:

- *Linear Regression*, where the dependent variable is predicted from one or more independent variables defining a hyperplane which best fits a cloud of points of a dataset, normally such that the sum of the squared errors of the distances between the data points and the hyperplane is minimal, as in:

$$y_k = w_k \cdot x_k + b, \tag{4.14a}$$

$$\arg\min_{w} ||y_k - (w_k \cdot x_k + b)||^2 \tag{4.14b}$$

  where x_k is the independent variable, y_k is the dependent variable, w_k are the model parameters and b is an error term.

- *Classification and Regression Trees (CART)*, where at each node a question concerning the independent variables is asked, splitting the dataset. At the leafs, the values of the dependent variable are stored.

- *Bayes-based approaches*, where the probability of a dependent variable C given an independent variable F is given as:

$$C_n = \arg\max_{C_n} P(C_n|F_k) = \arg\max_{C_n} P(C_n)P(F_k|C_n)) \tag{4.15}$$

  A specific case of a Bayes-based approach are the neural networks, as introduced in Section 2.4.

The relation between the semantics of the text and the acoustics of the expressiveness, intuitively, is highly nonlinear. The predictor model has to be sophisticated and capable of dealing with this nonlinearity. Neural networks are considered to be the most suitable approach. The concrete implementation for semantics-to-acoustics mapping with neural networks is described in section 4.3.2.

4.3 Experiments

This section describes the implementation of the prediction of acoustic features from semantics. The basic idea, as described above, is to use a regression model to predict the acoustic feature vector from a semantic one. The acoustic feature vector can then be used as a centroid, and the surrounding acoustic data can be used to form a cluster and use it to train speech synthesis models, similar to the approach in Chapter 3. The main difference is that the cluster centroid is predicted from semantics, which allows a series of applications. For instance, a text can be read with expressive voices where the expressiveness is adapted to each sentence in function of its semantic vector. Another application could be a search engine for emotional speech, where a semantic vector is calculated for an emotional key word. The predicted acoustic vector can then be used to form a data cluster and train an emotional voice. The next section introduces the proposed system framework, where these applications can be derived from.

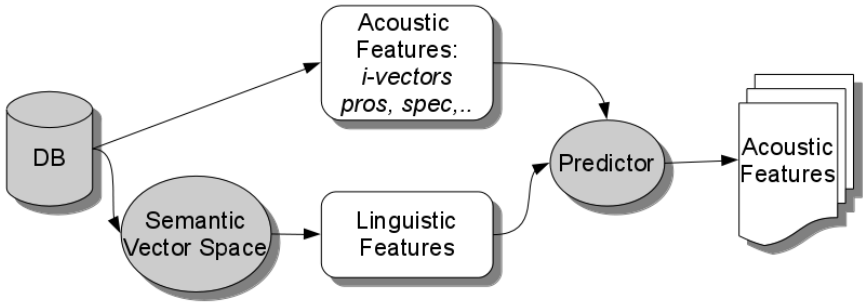


Figure 4.8: Framework of the proposed training approach.

4.3.1 Experimental framework

First, the semantic vector space is trained using the Skip-gram implementation in *word2vec*, as by Mikolov et al. (2013), on the Spanish portion of the *Wikicorpus*, as by Reese et al. (2010), with 120 million words. Then, a book is projected into the semantic vector space, such that for each book sentence a semantic vector is calculated. The book is also available as an audiobook, concretely exactly the same as used in the experiments in Chapter 3, of approximately 8.8 hours of duration and containing 7903 sentences. The semantic vector for each sentence is calculated as the middle point between the vectors of the words which constitute the sentence. As the book is also available as an audiobook, for the utterances of each sentence, an acoustic feature vector is extracted, the best performing feature vector from the experiments in Chapter 3. The acoustic feature vectors are composed of 600 dimensional i-vectors trained from MFCCs, 12 dimensional i-vectors trained from F0, and 8 dimensional vectors with syllable and silence statistics, 620 dimensions in total. The MFCCs and F0 features are extracted using *AHOCoder*, as by Erro et al. (2011a). The syllable and silence duration with *Ogmios* speech analysis tools, as by Bonafonte et al. (2006), and the i-vectors using the *Kaldi software*, as by Povey et al. (2011). A classifier is trained to predict the acoustic feature vectors from the semantic feature vectors. The whole process is illustrated in figure 4.8.[1] The classifier are different for the different experiments and will be specified in respective sections.

For the prediction of the acoustic feature vector, an input text is projected into the semantic vector space and the semantic vector is calculated for the input text. Next, it is fed into the classifier which predicts the acoustic feature vector, as illustrated in figure 4.9.

When the acoustic vector is predicted, it can be used as a centroid to form a data cluster and use it for voice training. In this work, speaker adaptation is

[1]Part of the proposed framework was developed at the University of Texas at El Paso under the supervision of Prof. Nigel Ward.

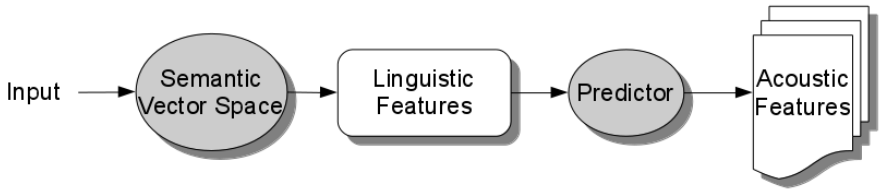


Figure 4.9: Framework of the proposed acoustic feature vector prediction.

performed on the cluster data to train synthetic expressive voices. The AVM is trained using the whole audiobook corpus. Further details will be explained in the respective experimental sections.

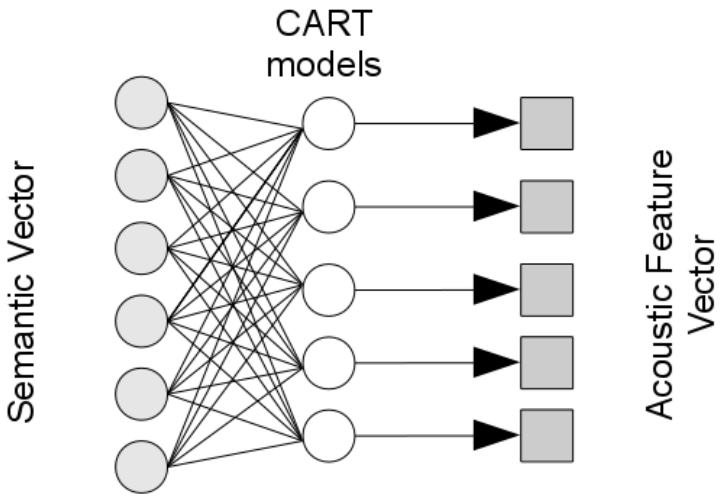
4.3.2 Predicting Acoustic Feature Vectors from Semantic Vectors: an Analysis

For the first experiment, as published by Jauk et al. (2016), first, the correspondent models were trained. The linguistic input feature vector is composed of three parts. The utterance in question, the left and the right contexts are projected into the SVSM and the coordinates are extracted, 1800 totally since the utterance and the context vectors have the length 600 each. The context on the left and on the right is composed of the next and of the preceding three words, respectively. The amount of words to take into account has been determined experimentally, the performance declining from the fourth word on. The reason might be that the context becomes too specific and moves away in the semantic space pushed by the words farther away from the sentence in question. Additionally, models were trained, which predict acoustic features not only from the semantic vectors, but also taking into account the acoustic feature vectors of the previous utterances, i.e. the acoustic context. In the prediction, the previously predicted feature vector is used for the next utterance, except for the first utterance which uses the acoustic corpus center as starting point.

Two predictor models are tested. The first one is a CART network. The CART network is designed such that each tree predicts one value of each dimension of the output vector. The input vector, in this case the semantic vector, is shared for all dimensions of the output vector, in this case the acoustic feature vector, as illustrated in Figure 4.10.

The second model is a bottleneck designed DNN model shown in Figure 4.11. The design was defined experimentally and choosing the one which yielded best prediction results. There are several intermediate (hidden) layers, and in between, Dropouts, as by Srivastava et al. (2014), of 0.5 are applied to lower any possible over-fitting effect. At the output of the network a *tanh* activation function is used, so the output features are normalized between $[-1, 1]$.Since the entrance layer has a rather larger number of neurons, the first hidden layer is

Figure 4.10: CART network design.



also relatively large (1024 in the case of only semantic coordinates, 1500 for the semantic and acoustic combination). The next layer shrinks down to 256 neurons. There are several hidden layers with this number of neurons, which is then increased to 512, and to 620 in the output layer. In the case of the semantic prediction best results were achieved with 10 hidden layers. In the case of the semantic and acoustic combination the number of hidden layers is 5.

Figure 4.11: DNN framework.

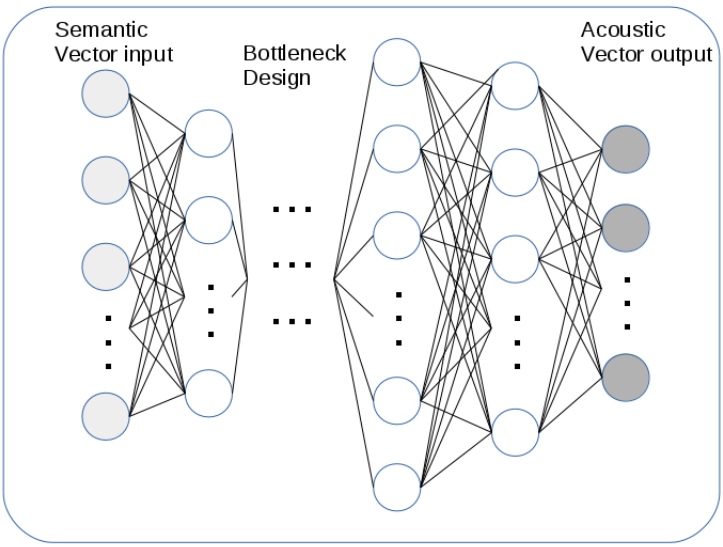


Table 4.3: Distance results. Means and variances of distances to the original acoustic feature vectors.

	CART.sem	CART.acu	DNN.sem	DNN.acu	rand
MEAN	2.44	2.42	1.89	1.89	2.69
VAR	0.51	0.37	0.38	0.38	0.51

Four excerpts from the audiobook were selected for the evaluation, a total of 106 utterances. The test set was excluded from training. All test utterances and their context were projected into the SVSM obtaining the semantic coordinates. Then acoustic coordinates were predicted from the semantics for each of the four experimental conditions: (1) using CART with only semantics; (2) CART combined with acoustics; (3) DNNs with semantics only and (4) DNNs combined with acoustics.

The predicted acoustic feature vectors were compared to the original feature vectors for the utterances extracted from the corpus measuring the Euclidean distance.

As a reference for the distance measure, the same test set was randomized and the distances were compared between the original and the shuffled vectors. Also, a closer analysis of distances between the original and the predicted vectors is conducted.

Results

Table 4.3 shows the results for the distance measures between the predicted feature vectors and the original feature vectors, in comparison to the distance of the original vectors to randomized original vectors. ANOVA is used to test the significance of the difference between these distances. Table 4.4 shows the ANOVA F-values for the distances.

DNNs have produced predictions that significantly differ from random. CART did a slightly better prediction including the left acoustic context in the predictor vectors, reflected in the distance variance. However, looking at the ANOVA F values, although the F values are higher than the critical F value, the p values for the predictions with CART are 0.003 and 0.006, for the predictions with only semantic vectors and including the acoustics, respectively. So, the CART prediction is probably not significantly better in comparison to the shuffled data.

Between CARTs and DNNs, there is a significant difference in performance. However, combining semantic and acoustic features for the prediction did not result in any significant improvement.

Figure 4.12 shows the distance plot of distances between the original acoustic vectors and the predicted vectors, for the four conditions, and the 106 utterances. The lower the line, the better is the prediction. The DNN predictions with semantics alone and with the combination with the acoustics are so similar that the lines practically overlap.

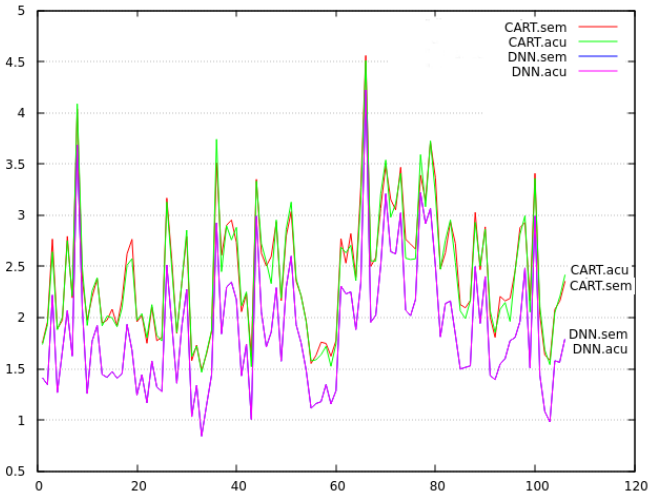
It can be observed that for some utterances the prediction is worse than for others. There are some peaks of larger distances, especially around the utter-

Table 4.4: ANOVA results between the four conditions and random. $\alpha = 0.05$, critical $F = 3.8861$. Values marked with * have a p value above 0.0025

	CART.sem	CART.acu	DNN.sem	DNN.acu
rand	7.852*	8.900*	76.897	76.908
CART.sem	-	0.044	43.551	43.561
CART.acu	-	-	40.520	40.530
DNN.sem	-	-	-	0.000

ances 8, 36, 63 and in the area between 66 and 80. The utterance 8 is just a *"yes"*, so there is not much expressive information encoded, the utterance 36 is a proper name, also difficult to relate to prominent expressiveness, at least without taking into account larger context or world knowledge. The utterance 63 just says *"exclaims"*, with also very little context (*"perfect"* on the left and *"to bring your"* on the right). The area between 66 and 80 is a conversation in very general terms, including phrases like *"yes, please"*, *"hello"*, *"he said"* and some more.

Figure 4.12: Euclidean distance plot of the predicted to the original distances for the 106 utterances.



Possibly, rather large utterances codify more expressive information than short ones. Anyway, it is clear that reasonable prediction is truly possible for a reasonable portion of utterances, and that the deep neural networks show better performance for given task.

4.3.3 Automatic Expressive Reading of Text

The second experiment evaluates the usability of the proposed prediction approach for expressive speech synthesis. Two tasks are designed for the evaluation. The first task includes the reading of a paragraph of the book which was

used to train the voices. This task as a topline (proof of concept) since the audiobook is known. Two classifiers are used for the prediction: a DNN predictor and a nearest-neighbor selection method. The paragraph was excluded from training of the DNN predictor (not applicable to the nearest-neighbor method). The paragraph is a dialogue between two book characters and narrator comments; it contains 16 sentences. Each sentence is projected into the semantic vector space and its coordinates are extracted. In the DNN prediction, acoustic coordinates are predicted for each sentence, which are then used as centroids to select In the alternative nearest-neighbor method, 50 nearest neighbors are selected directly from the acoustic space using the predicted acoustic feature vector as a centroid. In both cases, an individual voice is trained for each sentence of the input text. For a baseline comparison, the paragraph is synthesized using a neutral voice, trained from approx. 10 hours of studio recorded read speech. The participants are asked to rank the three voices by best expressive performance and by quality.

The second task is similar to the first task, but the paragraph to synthesize is extracted from a new book that has not been projected into the semantic vector space nor used for the DNN training. Though, to maintain the context, this book is the continuation of the first book. The DNN prediction is identical to the one in the first task: acoustic coordinates are predicted for each sentence, which are then used as centroids to select 50 nearest neighbors in the acoustic feature space. Here, in the alternative method, the semantic vector calculated for each sentence is used as centroid in the semantic vector space (opposed to the first task where the predicted acoustic vector was used as a centroid in the acoustic vector space). Then, 50 nearest neighbors are selected in the semantic vector space, and the corresponding acoustic feature vectors of the selected sentences are used to train the voices. Again, with both methods, for each sentence an individual expressive voice is trained. And also here, as a baseline, the paragraph is synthesized using the neutral voice.

Since whole paragraphs are synthesized for both tasks, context can be used to achieve better prediction. So, each predictor vector is composed of the semantic vector for the sentence in question, and of two additional vectors, for the left and for the right contexts respectively. These context vectors are calculated using three closest words on the left and three closest words on the right of the sentence in question, as in the first experiment.

Results

A total of 21 subjects participated in the experiment, some of them experienced with speech technology (either development or usage), and others not. Table 4.5 presents the results for the first two tasks. Participants had the option to prefer two synthesized paragraphs, or all of them, if sounded equally, again, regarding expressiveness and quality. In both tasks, regarding expressiveness, the results show clear preference of both synthesis methods over the neutral voice. In the first task, or the nearest-neighbor method was chosen to be better, or at least equal to the DNN based method. In the second task, almost half of the subjects chose the nearest-neighbor method and the DNN based method equally good. If not, the DNN method was slightly preferred. In synthesis quality, there was no significant preference for none of the voices.

Table 4.5: Prediction method preferences by users for the first two tasks. DNN method, nearest neighbor (NN) method, neutral voice.

	DNN	NN	neutral	DNN =NN	NN =neutral
Task_1	0.19	0.43	0.0	0.38	0.0
Task_2	0.29	0.14	0.04	0.48	0.05

4.3.4 Creating *adhoc* Expressive Voices

This task evaluates the system as a search engine for emotional training data. Key words or phrases, which semantically represent the emotion of the desired synthetic voice, are introduced into the system. Three emotional voices have been trained: *happy*, from key word "happy", *angry*, from key phrase "I don't want!" and *suspense*, from key phrase "Mysterious secret in silent obscurity." The keywords are projected into the semantic vector space obtaining an embeddings for each of the keywords. 20 sentences, which are nearest neighbors to the predicted embeddings in the semantic vector space, are selected from the audiobook. Since this task is rather a data search task with the goal to train emotional voices, rather than automatic reading, a minimal manual intervention was allowed. The best sentence is selected manually from the 20 sentences proposed by the system. The criterion for the selection is the individual impression of suitability of the sentence to represent the desired emotion. Then, the acoustic feature vector extracted from the selected candidate is used to form a data cluster of 50 nearest neighbors and is used for voice adaptation. Although there is a manual selection involved, the effort is negligible in comparison to labeling of a corpus.

Using the three expressive voices and the neutral voice, seven sentences are synthesized. The sentences are designed to reflect semantically the emotion of the voices. The sentences are listed below, translated from Spanish.

- **Happy_1:** We have won the paella competition.
- **Happy_2:** Finally, the holidays begin.
- **Angry_1:** You are an idiot. Never speak to me again.
- **Angry_2:** You are a goof-off. If you don't push yourself we won't win anything.
- **Suspense:** In the middle of the night, a silent shadow moved along the corridor.
- **Sad:** We haven't won the paella competition.
- **Neutral:** In many civilizations seven-days weeks are in use.

As seen in the sentence list, there is a *sad* sentence, but there is no sad voice. This is because it is hypothesized that the *suspense* voice can also be used for *sad* or even *neutral* content. This might be also true for the *neutral* voice.

Results

Table 4.6: Task 3. Voice preference by users for each sentence.

	happy	angry	suspense	neutral
Happy_1	**0.29**	**0.38**	**0.24**	0.10
Happy_2	**0.52**	0.24	0.10	0.14
Angry_1	0.14	**0.48**	0.24	0.14
Angry_2	**0.38**	**0.43**	0.14	0.05
Suspense	0.0	0.05	**0.81**	0.14
Sadness	0.19	0.05	**0.43**	**0.43**
Neutral	0.10	0.05	**0.43**	**0.43**

A total of 21 subjects participated in the listening test, some of them experienced with speech technology (either development or usage), and others not. The task was to choose the most suitable out of the four voices: *happy*, *angry*, *suspense* and *neutral*, for each of the seven sentences presented above. Table 4.6 shows the preferences of the test for the adhoc voices. The first *happy* sentence is divided between the three expressive voices, with a slight preference of the *angry* voice. For the second *happy* sentence, there is a clear preference of the *happy* voice. A possible explanation for given distribution for the first *happy* sentence is that it might be ambiguous to the listeners. However, the *happy* and the *angry* voices both sound similar and can be appropriate to both types of sentences. In fact, the *angry* voice does not sound really angry, it is more "book" angry, and meant for children.

For the first *angry* sentence there is a clear preference for the *angry* voice, with the *happy* voice on the second place. The second *angry* sentence is divided between the *happy* and the *angry* voice.

For the *suspense* sentence there is a very clear preference of the *suspense* voice. The *sad* and *neutral* sentences are also divided between the *suspense* and the *neutral* voice. There is no explicit *sad* voice, however the *suspense* voice does a good job imitating sadness, and as the results show, also the *neutral* voice.

4.4 Discussion

This chapter has summarized the prediction of relevant acoustic information from semantic vector embeddings extracted from plain text. First, different methods of calculating the embeddings have been presented, ranging from simple bag-of-words models to vectors extracted with neural networks. Then, it has been explained, how these embeddings can be used as input for regression models in order to predict acoustic feature vectors. Finally, experiments with different applications have been described and the results have been presented.

Vector embeddings are an efficient tool to describe semantic (and syntactic) phenomena. As have been shown, they also can be used to predict expressiveness, or rather expressiveness related acoustic features, from plain text. The quality of the prediction rises with sentence length. Also, as Socher et al. (2013) argue, talking about the sentiment training database, the authors mention that longer

sentences tend to be positive or negative, while shorter ones tend to be neutral. However, in real dialogue situations expressions are communicated in all type of utterances, where short utterances can be very expressive (for instance very determined *Yes!* or *No!*). Pennebaker (2011) states that in speech, expressiveness is often communicated in semantically rather secondary words such as pronouns.

Regarding automatic paragraph reading, although the tasks were completed successfully, there is still a lot of room to improvement. Context is surely very important for the prediction, and the way how it is handled. On the other hand, the embeddings by themselves should be designed to better represent expressiveness than semantics.

For the part of the emotional search and voice training, the proposed approach turned out to be very effective. It would be interesting to evaluate its effectiveness on larger and more "real-life" like data, maybe interviews, and also expanding it to other types of expressiveness, not only emotions.

Normally, the effectiveness of genuine semantic vectors relies on the meaning which can be extracted from word structure, word combinations, word contexts etc. However, expressiveness only partly can be extracted from pure semantics, here the pragmatic, i.e. the situational meaning is much more important. The Standford sentiment parser does predict pragmatically relevant embeddings which are strongly correlated with prosody and expressiveness of the sentence, as will be discussed in Chapter 5. However, the training data for the sentiment parser is manually labeled and not automatically derived from text. The automatic pragmatic prediction from text remains an open issue in expressive speech synthesis. Additionally, the Stanford parser is trained on written information only, such that there is no sure connection between the sentiment of a written sentence and the expressiveness of the same sentence if it should actually be spoken.

Another discussion topic is the HMM based synthesis method, which produces regular quality speech. The problem with the HMM based synthesis, among others, is the clustering of speech. Since the clustering is performed using syntactic and acoustic criteria, it does not guarantee optimal clusters of expressive speech, including data which should be outside of the cluster and excluding data which should be inside. This rough and non-optimal separation contributes to artifacts and lowers synthetic speech quality. A more effective method are the neural networks, which use all the training data to optimize synthetic output. A first approach to expressive speech synthesis using neural networks is presented in the next chapter.

Chapter 5

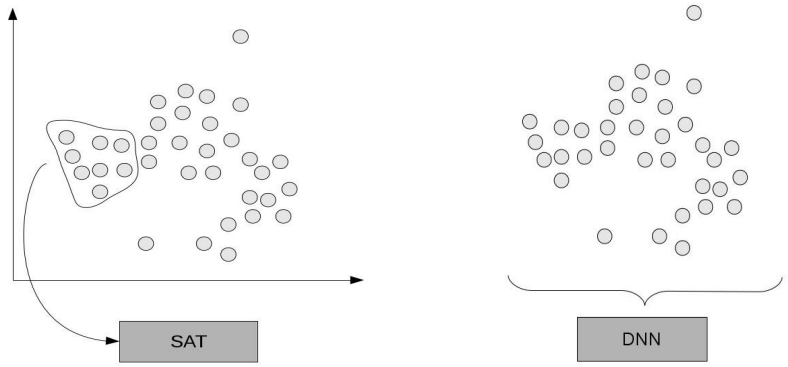
NN-based expressive speech synthesis with sentiment embeddings

In the previous study, semantic vector representations of text have been used to perform a look-up in the training corpus for expressive speech data according to the textual input, such that, relying on semantic information, data clusters were used to train expressive voices via speaker adaptation. A logical evolution of this study is to use embeddings which are more dedicated to the expressiveness in text. The earlier introduced Stanford sentiment parser is such a tool, which provides vector embeddings reflecting the sentiment, i.e. the positiveness or the negativeness of the text. For more details refer to Section 4.1.4.

The Stanford parser is trained on labeled movie reviews, originally collected and published by Pang and Lee (2005). The input to the Stanford parser is a textual unit, word level or more. First, the input is parsed and converted into a binary tree structure. Then, for each level the system predicts a sentiment. The format can be just a value, between *positive*, *negative* or *neutral*, a probability of belonging to one of the five categories *very positive*, *positive*, *very negative*, *negative* or *neutral*, or a vector embedding in a sentiment vector space.

A further improvement is the migration from HMM-based synthesis to DNN-based synthesis. A main drawback of the HMM-based synthesis is that the training data is clustered. This is a disadvantage, for clustering relies on extracted features, in this case representing expressiveness, however, even if the features are very good, there will always be an error. This will cause that certain data points which should belong to the training cluster are not inside, and certain other, which do not belong to the training cluster, are inside.

DNN-based synthesis, in certain manner, avoids this problem because the network sees the complete data set, and the neurons "decide" according to the training criterion, which output data (speech), corresponds to which input data (in this case, embeddings). In this sense, there is a kind of abstracted intern clustering optimized according to the training criterion. Both concepts are shown in Figure 5.1.



(a) SAT training from data clusters. (b) DNN training from the whole data.

Figure 5.1: Training from data SAT vs DNN

In previous work, neural network based systems have already been combined with semantic vector input, though not for expressive speech. To name a few, Wang et al. (2015) use word embeddings to substitute TOBI and POS tags in RNN-based synthesis achieving significant system improvement. Wang et al. (2016) enhance the input to NN-based systems with continuous word embeddings, and also try to substitute the conventional linguistic input by the word embeddings. They do not achieve performance improvement, however, when they use phrase embeddings combined with phonetic context, they do achieve significant improvement in a DNN-based system. Wang et al. (2016) enhances word vectors with prosodic information, i.e. updates them, achieving significant improvements.

In comparison to these systems, the system proposed here uses sentiment embeddings, i.e. the embeddings have an expressive meaning. Some speech synthesis systems have already used sentiment information. For instance, Trilla and Alias (2013) already used sentiment analysis on sentence level for an expressive TTS. Vanmassenhove et al. (2016) also used sentiment combined with emotion labels for an HMM-based system. Sudhakar and Bensraj (2014) implemented a TTS in Matlab which used sentiment information trained with fuzzy neural networks evaluated in a news domain. Differently from these systems, the proposed system uses sentiment for a DNN-based TTS in an audiobook domain, which is considerably open and rich in expressive speech.

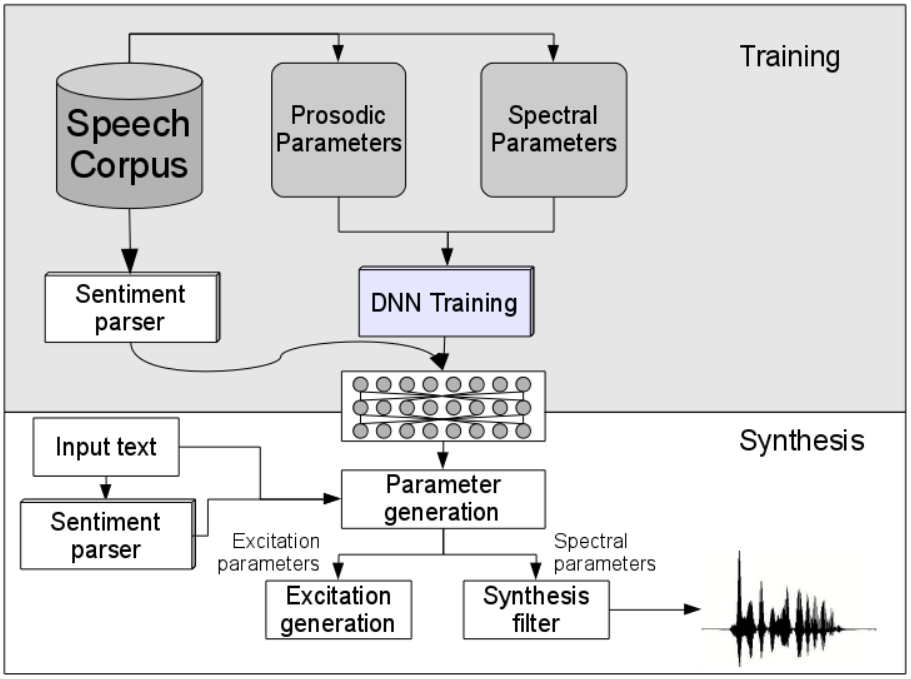
Section 5.1 describes the system architecture of the DNN-based synthesis with sentiment embeddings. Section 5.2 presents objective statistics measured on pitch for sentiment-analyzed corpora. Section 5.3 presents a preliminary experiment, and section 5.4 presents the main experiment with corresponding results, following by a discussion in Section 5.5.

5.1 System architecture

The proposed architecture is basically an extension of the system presented in figure 2.12, where the DNN receives an additional input, the sentiment vectors,

as shown in figure 5.2.[1]

Figure 5.2: Proposed DNN system architecture using sentiment embeddings.



The underlying DNN system has the following specifications, as by Takaki and Yamagishi (2016). For each utterance, a 60 dimensional MFCC vector, log F0, 25 dimensional band aperiodicity measures, and for each, dynamic and acceleration features are extracted. The log F0 is linearly interpolated and voiced/unvoiced marks are used as parameters. *Combilex*, by Richmond et al. (2009), is used to create context label files. First, an HMM-based training is performed, estimating phoneme boundaries. Then, the deep neural network is trained. The DNN is implemented with 5 hidden layers, each containing 1024 neurons, minibatch size of 256 and using Adagrad gradient optimization. *Straight* vocoder, as by Kawahara et al. (1999), is used to generate the waveform.

As proposed, an additional linguistic input is introduced, the sentiment predicted by the Stanford sentiment parser. Here, different input combinations are tested. Probability and embeddings are used alternatively in following configurations:

- **Sentence level (sl)**: Probabilities and embeddings on sentence level are added to the input.
- **Word level (wl)**: Word level probabilities and embeddings are used.

[1]The system was developed at the National Institute of Informatics (NII) in Tokyo and the proposed implementation was realized in cooperation with NII under supervision of Prof. Junichi Yamagishi, in the mark of the NII International Internship Program.

- **Word context and tree distance (wcd)**: Word context includes word level embeddings with two word embeddings on the left and on the right of the current word. It also includes the hierarchical tree distance for each word, i.e. the distance measured in number of tree nodes which separate two words.
- **Sentence level, word context and tree distance (swcd)**: Here, the above condition is combined with the sentence level embedding with the aim to stabilize the overall utterance prosody.

To visualize the input vectors, the probability vectors are composed as follows: S

$$P = [p_{vneg}, p_{neg}, p_{neu}, p_{pos}, p_{vpos}] \tag{5.1}$$

where p_{vneg} is the probability of the category *very negative*, p_{neg} the probability of the category *negative*, etc. The probability vectors are provided on sentence level (sl) and word level (wl), in the respective cases. When word context was taken into account, probability vector of the word in question and the probability vectors of two words on the left and two words on the right were used. Also the tree distance, which is the hierarchical distance counted in the number of binary tree nodes between words is added, such that the input vector for each word for the system (v_wcd) is composed as follows:

$$P = \{P_{l_2}, P_{l_1}, P_c, P_{r_1}, P_{r_2}, D_t\} \tag{5.2}$$

where P_c is the probability vector for the current word, the P_{l_2} is the probability vector for the second word on the left, P_{l_1} is probability vector for the first word on the left, P_{r_1} is probability vector for the first word on the right and P_{r_2} is the probability vector for the second word on the right, each of the probability vectors as defined in equation 5.1. D is the hierarchical tree distance.

In the case where instead of the probabilities the semantic embedding from the sentient vector space was used directly, the composition is identical to the (v_wcd) except that instead of probability vector, the vector embeddings were used, each of dimension 25. On tehcnical side, the vectors were always inserted on frame level. So for instance, when sentence level probabilities were used, for each frame an additional vector of five probabilities was added, which is equal for each frame of the sentence. When using word level probabilities, the additional vector changed on word boundaries.

In order to better understand the influence of the sentiment information, objective measures on prosody are performed. The results are presented in the Section 5.2.

5.2 Objective test

The objective test consists of two parts. First, pitch statistics are calculated for two corpora in function of the sentiment category predicted by the Stanford parser for each of the sentences of the corpora. The pitch is extracted using the *pitch* tool from the SPTK tookit. Second, the DNN system is trained using

different sentiment inputs, among the combinations described above, and the predicted pitch of the systems trained with sentiment vectors is compared to the predicted pitch trained without sentiment vectors, showing on examples how the DNN system learns from the sentiment input.

For the first part, two corpora are used: (1) a corpus of neutral speech (c1), containing 1000 sentences of a total duration of approximately 1 hour, read by a professional female speaker of American English; (2) a clean portion of an audiobook (c2), containing 5039 sentences of a total duration of approximately 5 hours, read by a semiprofessional male speaker of American English. For each sentence of both corpora, sentiment category is calculated using the Stanford parser. From corresponding utterance, F0 is calculated, including means, variance, range, range means, and range variance.

The numbers of sentences belonging to each sentiment category for the neutral corpus (c1) are listed in Table 5.1. The extreme points *very positive* and *very negative* are very underrepresented. Since this is a neutral corpus, the speaker probably was instructed not to reflect expressively the content of the sentences.

Table 5.1: Number of sentences for each sentiment category for the neutral speech corpus c1.

category	#sentences
very negative	2
negative	517
neutral	279
positive	198
very positive	4

Table 5.2: Pitch statistics in function of sentiment for the neutral speech corpus c1. Listed are mean, variance, range, range-mean and range-variance.

	mean	var	range	r-mean	r-var
very negative	261	67	432	252	20
negative	271	55	429	228	34
neutral	264	66	502	248	36
positive	266	65	490	240	38
very positive	264	66	539	244	33

Table 5.2 shows the statistics for the neutral corpus c1. In general, the values do not show significant changes except for the pitch range, which seems to be higher for the neutral and positive categories in comparison to the negative ones. It has to be taken into account, that the two extreme categories, very negative/positive, are very underrepresented, and thus can not be relevant for the statistics.

The numbers of sentences belonging to each sentiment category for the audiobook corpus (c2) are listed in the table 5.3. Also here, the extreme categories

are underrepresented. It has to be noted that the audiobook speaker in general is not too expressive. On the other hand, there is no reflection of the acoustic expressiveness in the sentiment values since these are predicted from text only. Examining the sentences, the prediction is not always perfect, yielding many "erroneous" sentiment categories. This means that often the category predicted by the parser is not the category one would intuitively assign to a sentence. For instance, the sentence *"Jim shook his head and said:"* is labeled as positive, though intuitively it is rather neutral, also in the interpretation by the reader it is rather neutral. Another example, *"Nothing less than a great magnificent inspiration!"* is labeled as negative, though intuitively it is rather positive, and also the reader interpretation is rather positive. These are only two examples, however, there a many more of this kind in the audiobook.

Table 5.3: Number of sentences for each sentiment category for the audiobook corpus c2.

category	#sentences
very negative	7
negative	1901
neutral	2389
positive	721
very positive	21

Table 5.4: Pitch statistics in function of sentiment for the audiobook corpus c2.

	mean	var	range	r-mean	r-var
very negative	105	24	224	113	33
negative	120	29	431	125	45
neutral	121	28	734	108	57
positive	122	30	399	122	46
very positive	135	36	355	135	51

Table 5.4 shows the statistics for the audiobook corpus c2. Also here, the extreme categories are poorly represented, although better than in the neutral corpus. The neutral category has the largest range, but the smallest mean range. Apart of this, there is a slight tendency of F0 to increase, when categories are positive in comparison to the negative ones.

In order to better understand how neural networks learn with the additional sentiment information, some example sentences will be visualized. The graphics show the sentiment category (purple line), the probability of that category (blue line), the predicted F0 curve (green line), and the predicted F0 curve without sentiment information (red dashed line), for each word in the sentence. The categories plotted with index numbers between 0 (=very negative) and 4 (=very positive).

The first visualization is for the sentence *"My house is green with a big yellow door."*, categorized as *neutral* at the sentence level, with the probability of

0.8946. The individual categories and probabilities for the words are listed in the table 5.5. All words are categorized as neutral with very high probabilities.

Table 5.5: Word categories and probabilities for the neutral sentence.

word	category	probability
my	*neutral*	0.9974
house	*neutral*	0.9848
is	*neutral*	0.9894
green	*neutral*	0.9689
with	*neutral*	0.9917
a	*neutral*	0.9902
big	*neutral*	0.9947
yellow	*neutral*	0.9552
door	*neutral*	0.9812

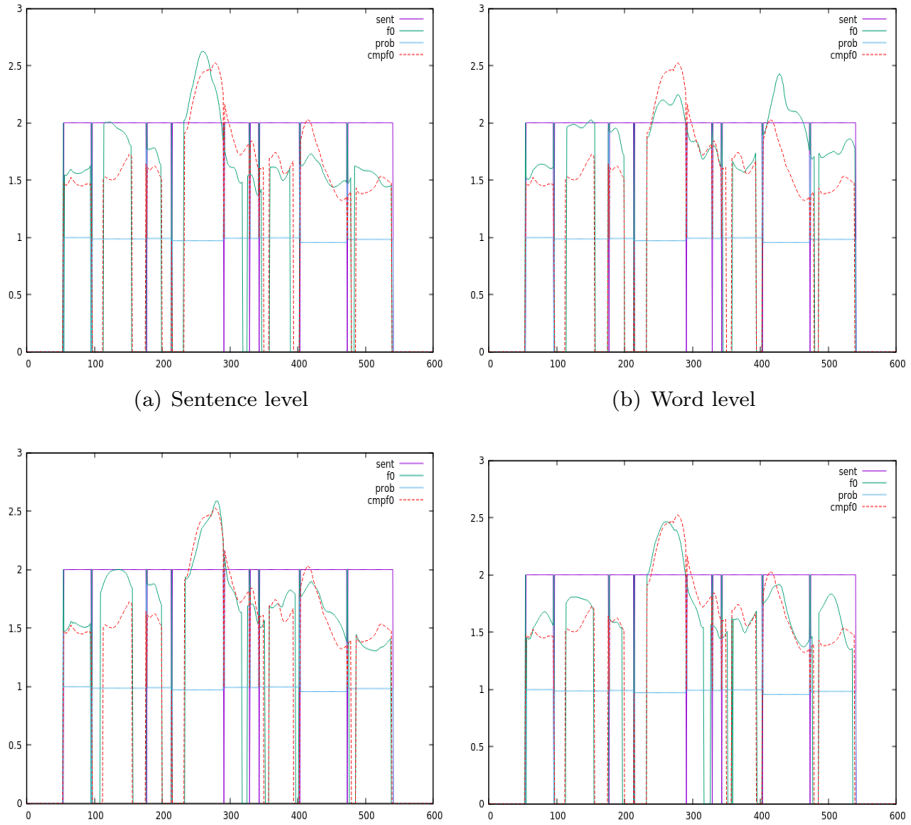
As can be seen in Figure 5.3, for the neutral sentence, the general F0 shape is similar between the models. There are slight differences in accentuations though. In all versions, there is a stronger accent in the beginning, on the word "house". For the word level, the strongest sentence accent, on the word green, is lower. And there is a stronger accent on the first syllable of the word "yellow". For sentence level, word context and tree distance, there is a stronger accent on the word "door".

Table 5.6: Word categories and probabilities for the negative sentence.

word	category	probability
the	*neutral*	0.9933
awful	*neutral*	0.6307
soundtrack	*neutral*	0.9915
was	*neutral*	0.9960
disgusting	*negative*	0.5070
and	*neutral*	0.9960
made	*neutral*	0.9992
me	*positive*	0.9462
puke	*neutral*	0.7497

The next sentence is *"The awful soundtrack was disgusting and made me puke."*, categorized as *negative* at the sentence level, with a probability of 0.6254. The individual categories and probabilities for the words are listed in the table 5.6. Most of the words are categorized as neutral with high probabilities, surprisingly also the words "awful" and "puke", although with lower probabilities. Intuitively, these two words would be negative. Also unexpected, the word "me" is categorized as positive. The only word which is categorized as negative, is "disgusting" and apparently determines the rest of the classification.

Figure 5.4 shows the plots for the negative sentence. Also here, the general shape of the F0 curve is similar, but there are slight differences. For instance,



(a) Sentence level

(b) Word level

(c) Word context and tree distance

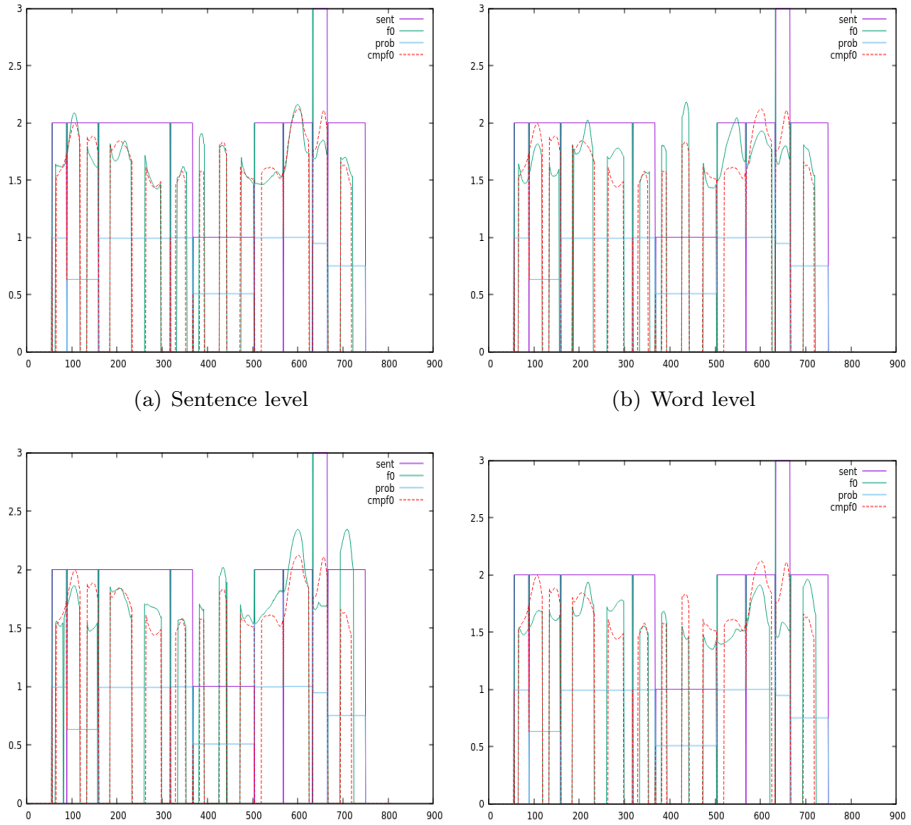
(d) Sentence level, word context and tree distance

Figure 5.3: Pitch visualization for the neutral sentence. For each figure: *sent*, purple line, is the predicted sentiment category on word level; *prob*, blue line, is the probability of that category; *f0*, green line, is the predicted F0 curve with the sentiment; and *cmpf0*, red dashed line, is the predicted F0 curve without sentiment.

the systems with word level and word context and tree distance information make a stronger accent on the word "disgusting". The word level system also accentuates the word "made". The two systems with the word context and tree distance makes a stronger accent at the end on the word "puke".

The next sentence is *"A woman's hair is wonderful."*, categorized as *positive* at the sentence level, with a probability of 0.5699. The individual categories and probabilities for the words are listed in the table 5.7. "Wonderful" is the only word categorized as positive, the other words are neutral. Also "'s" in "woman's" is interpreted as a word.

Figure 5.5 shows the plots for the positive sentence. Here, the sentence level, and especially the word level systems have accentuated the word "wonderful", which is the positive one. On contrary, the word context and tree distance systems have accentuated the word "hair".



(a) Sentence level

(b) Word level

(c) Word context and tree distance

(d) Sentence level, word context and tree distance

Figure 5.4: Pitch visualization for the negative sentence. For each figure: *sent*, purple line, is the predicted sentiment category on word level; *prob*, blue line, is the probability of that category; *f0*, green line, is the predicted F0 curve with the sentiment; and *cmpf0*, red dashed line, is the predicted F0 curve without sentiment.

The last sentence is *"I was extremely happy with the movie."*, categorized as *very positive* at the sentence level, with a probability of 0.6221. The individual categories and probabilities for the words are listed in the table 5.8. The word "happy" is categorized as very positive, the others as neutral.

Figure 5.6 shows the plots for the very positive sentence. Here, the most positive word, "happy", is not accentuated. The strongest accent lies in "extremely" which is categorized as neutral. The sentence and word level systems put a stronger accent on "I" at the beginning, and shallow the accent on "extremely". The other two systems reproduce the F0 shape without sentiment.

These examples show that there is a clear change in pitch prediction caused by the sentiment probabilities which are provided as an additional input to the DNN. Also, depending on the configuration of this additional input, the changes are different. It is clear that the sentiment parser is not always perfect and the

Table 5.7: Word categories and probabilities for the positive sentence.

word	category	probability
a	*neutral*	0.9951
woman	*neutral*	0.9957
's	*neutral*	0.9945
hair	*neutral*	0.9939
is	*neutral*	0.9894
wonderful	*positive*	0.8968

Table 5.8: Word categories and probabilities for the very positive sentence.

word	category	probability
I	*neutral*	0.9962
was	*neutral*	0.9960
extremely	*neutral*	0.9787
happy	*very positive*	0.7602
with	*neutral*	0.9917
the	*neutral*	0.9941
movie	*neutral*	0.9984

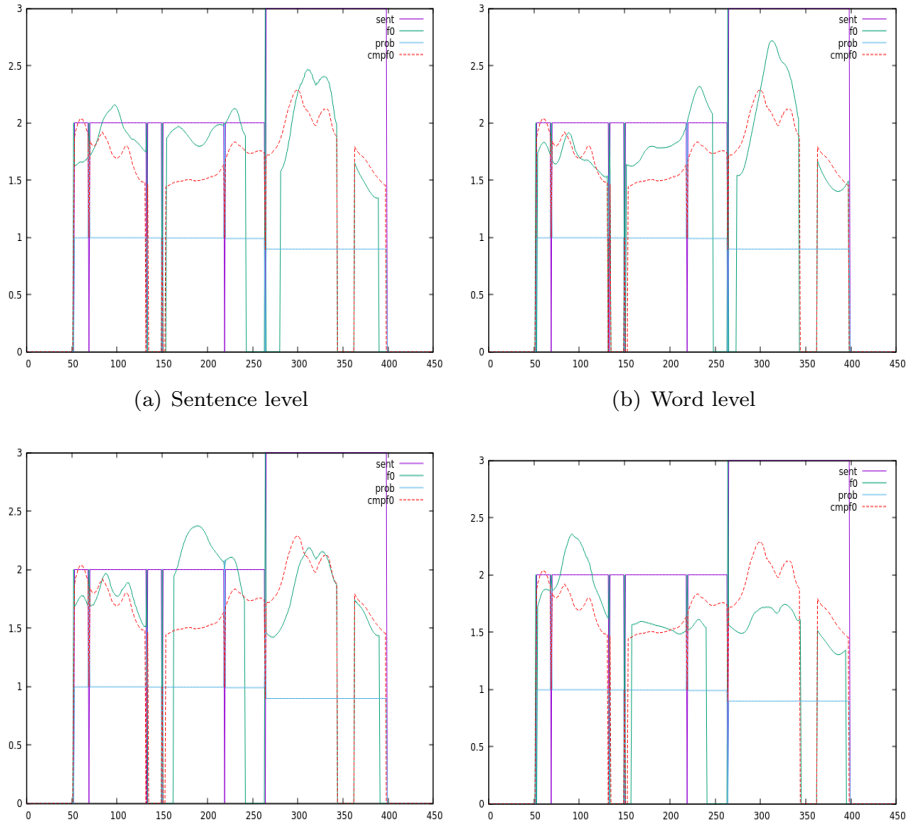
interpretation it makes is not always the intuited one. For instance, it was never possible to get a categorization "very negative" among the test sentences.

In order to verify the effect of the sentiment on the DNN, some input probabilities have been modified manually, introducing very high probabilities of, for instance, 5, for positiveness. This yielded a very high overall pitch of the voice, driving it to an unnatural sounding when the probability rose. On the other hand, raising the probability for the negative sentiment had the contrary effect, driving the pitch down, at some point making it sound very aspirated.

5.3 Preliminary experiment for choosing the best DNN-sentiment architecture

The first experiment has a preliminary nature and aims at selecting the best input configurations for the DNN system. For this, different types of outputs from the sentiment parser were introduced as additional input to the DNN system. The possible outputs of the parser were, sentiment category, sentiment probability and sentiment vector embedding in the sentiment vector space. Some of these sentiment outputs were combined and extended to use them as input for the DNN system, as follows.

Using sentiment probabilities, the following inputs have been introduced to the DNN system: *sentence level* (sl), *word level* (wl), *word context and tree distance* (wcd), *sentence level plus word context and tree distance* (swcd). Using sentiment vector embeddings instead of probabilities, only *word context and tree*



(a) Sentence level

(b) Word level

(c) Word context and tree distance

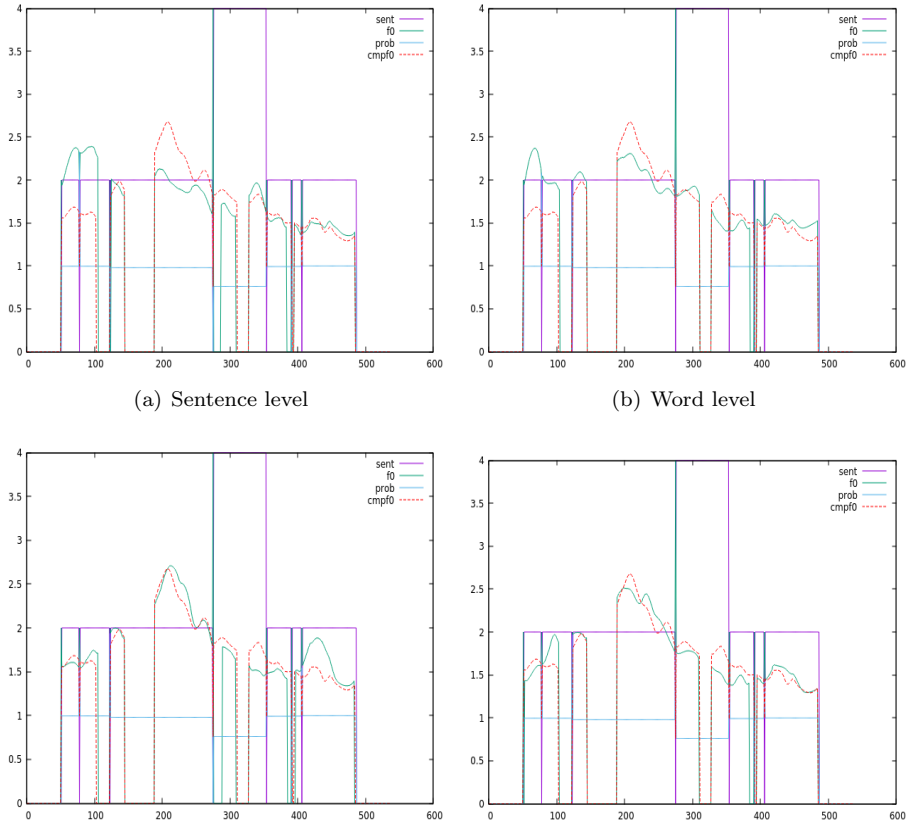
(d) Sentence level, word context and tree distance

Figure 5.5: Pitch visualization for the positive sentence. For each figure: *sent*, purple line, is the predicted sentiment category on word level; *prob*, blue line, is the probability of that category; *f0*, green line, is the predicted F0 curve with the sentiment; and *cmpf0*, red dashed line, is the predicted F0 curve without sentiment.

distance (v_wcd) has been tested. Also, a model without sentiment information (ws) has been trained.

The system, with the architecture as shown in Figure 5.2, and specifications as stated in Section 5.1, was trained with a clean portion of an audiobook corpus read by a semiprofessional male reader of American English. The audiobook portion contains 5039 sentences and is approximately 5 hours long. Apart of the features extracted for the DNN system, as stated in Section 5.1, for each of the sentence, a sentiment embedding and probability vector was calculated, using the Stanford sentiment parser, and added in the combinations described above as an additional input to the system on frame level (except for the case without sentiment).

Six test sentences, which were excluded from training, were synthesized using each of the six models, resulting in a total of 36 test samples. The synthesized



(a) Sentence level

(b) Word level

(c) Word context and tree distance

(d) Sentence level, word context and tree distance

Figure 5.6: Pitch visualization for the very positive sentence. For each figure: *sent*, purple line, is the predicted sentiment category on word level; *prob*, blue line, is the probability of that category; *f0*, green line, is the predicted F0 curve with the sentiment; and *cmpf0*, red dashed line, is the predicted F0 curve without sentiment.

sentences are listed in table 5.9.

It was indicated to the participants if the sentence was positive or negative to facilitate the task. No neutral sentences were included. The sentiment of

Table 5.9: Synthesized sentences for the preliminary experiment.

s1	*There was the very chance for another big killing.*	(neg)
s2	*For several minutes it had been lying quite motionless.*	(neg)
s3	*He too was 17 years old when he died.*	(neg)
s4	*He had fought because he was born a fighter.*	(pos)
s5	*I think Mr Duncan Smith has a very good record.*	(pos)
s6	*I found them to be a really lovely family.*	(pos)

the sentence was determined by the Stanford parser. The participants had to rate each system between 1 and 6, where 1 was very good and 6 very bad. The participants could also give the same rank to different systems if they considered that they performed equally.

5.3.1 Perceptual results

Since this experiment is designed as a preliminary test to preselect the best systems, in total, only 5 persons participated, 4 of them experts on speech technology, and 1 not. None of them were native English speakers. Table 5.10 shows the means and variances of the ranking for each of the systems. The systems which performed best are the *word level* and the *word context and tree distance* system using probabilities. The *word level* system has higher variance though. Bad performances had the systems with *sentence level*, *word context and tree distance* with vector embeddings, and the system without sentiment, this latter one never obtained the rank 1. No test statistics will be performed doubting the significance of the statistics since there are only few participants, furthermore the test is considered preliminary.

Table 5.10: System preferences. sl: sentence level, swcd: sentence level, word context and tree distance, wcd: word context and tree distance, wl: word level, ws: without sentiment, v_wcd: vector word context and tree distance

	sl	swcd	wcd	wl	ws	v_wcd
mean	3.57	4.07	2.77	2.8	3.97	3.5
variance	3.22	3.17	2.05	3.68	2.17	2.12

Table 5.11 shows the preferences for the categories positive and negative. Here can be seen, that the systems do not perform equally well for positive and negative sentences. The generally best performing systems are good at positive sentences, but not so good at negative ones. The *word level* system has such a high general variance because it has a high variance with negative sentences. This can be due to the fact that, the word level system is the one with the most variance of the F0, which is understandable because the word probabilities drive it. As also could be seen in Section 5.2, this means that the accentuations are stronger and are more varied, which can be perceived as good or bad for negative sentences. With other words, negative sentences can sometimes sound better with stronger accentuations, and sometimes without. For instance, *"This was disgusting!"* probably would sound better with a stronger accentuation, but *"My grandmother died."* would probably be rather plain. This point has also been commented by some of the participants. The best performing system for negative sentences is the *sentence level* system, which makes more sense since it probably has a more equilibrating effect.

Table 5.12 shows, for each system, the percentages of obtained ranking of 2 and above, 3 and above, between 3 and 4, 4 and below, and 5 and below. Here, the system which obtained highest percentages of best ranking is the *word level* system, while the *word context and tree distance* system is the one

Table 5.11: System preferences for positive and negative sentences. sl: sentence level, swcd: sentence level, word context and tree distance, wcd: word context and tree distance, wl: word level, ws: without sentiment, v_wcd: vector word context and tree distance

	sl	swcd	wcd	wl	ws	v_wcd
positive mean	4.33	4.53	2.33	2.33	3.67	3.4
positive variance	2.24	2.55	2.38	2.24	2.67	1.54
negative mean	2.8	3.6	3.2	3.27	4.27	3.6
negative variance	3.17	3.54	1.46	4.92	1.64	1.64

with the lowest percentage of bad rankings. In general, the usage of sentiment probabilities seems to be of advantage, if it is well applied.

Table 5.12: System rank ranges in parts per one, sl: sentence level, swcd: sentence level, word context and tree distance, wcd: word context and tree distance, wl: word level, ws: without sentiment, v_wcd: vector word context and tree distance

	sl	swcd	wcd	wl	ws	v_wcd
*above*2	0.37	0.2	0.47	0.6	0.23	0.2
*above*3	0.43	0.47	0.6	0.63	0.4	0.6
$3-4$	0.3	0.3	0.47	0.17	0.37	0.5
*below*4	0.57	0.53	0.4	0.37	0.6	0.4
*below*5	0.33	0.5	0.07	0.23	0.4	0.3

5.4 Main listening test for the DNN-sentiment evaluation

For the second experiment, the two best performing systems, *word level* and *word context and tree distance*, were chosen to synthesize 12 sentences in comparison to the system without sentiment, a total of 36 samples. The systems are the same used in the previous experiment, without any modifications. The synthesized sentences are listed in Table 5.13.

Also here, the sentiment was determined by the Stanford parser. The participants had no information whether a sentence is supposed to be positive or negative, they had to intuit it from the semantics. The task, as before, was to rate the systems, between 1 and 3, being 1 the best option and 3 the worst. The participants could rate the systems equally, if they considered them to be equally good or bad. They also had the option to disqualify a system, if they thought that it was not adequate for a sentence at all.

5.4.1 Perceptual results

A total of 20 persons participated in the experiment, 12 of them reported to be experts in speech technology development, two have experience as users with speech technology, the other do not have experience with speech technology, one of them was native US-English speaker. Table 5.14 shows the average rankings and variances for the systems. As can be seen, the best performing system is the *word level* system, however, with a high variance. The system without sentiment was disqualified 1 time, the word level system 3 times, and the *word context and tree distance* system 0 times.

Table 5.15 shows the P-values for one- and two-tailed t-tests with $\alpha = 0.05$. The tests show that there is a significant difference between the system without sentiment and the *word level* system, but no significant difference between the system without sentiment and the *word context and tree distance* (although it is close), nor between the *word level* and the *word context and tree distance*.

Table 5.16 shows the preferences divided by the sentiment. For positive and negative sentences, the *word level* system performed best, although for negative sentences with high variance. For neutral sentences, the *word context and tree distance* system performed best. Possibly it is due to the fact that it probably has an equilibrating effect.

Table 5.17 shows the P-values for the t-tests for negative, neutral and positive sentences. For negative sentences, there is a significant difference between the system without sentiment and the *word level* system, and no significant difference for the other systems. For neutral sentences, there is a significant difference between the system without sentiment and the *word context and tree distance* system, but not for the other systems. For positive sentences, there is only significant difference for the one-tailed t-test between the system without sentiment and the *word level* system.

Among the comments of the participants, several stated that in some cases it was difficult to decide which system was better. Looking at the results of the only native speaker, he prefers the *word level* system with an average rank of 1.64, and he mostly discards the *word context and tree distance* system, with an average rank of 2.42. Only for neutral sentences he prefers the system without

Table 5.13: Synthesized sentences for the main experiment.

s1	*And if you fail I will kill you.*	(neg)
s2	*I indicated that dreadful lee shore.*	(neg)
s3	*I exclaimed startled out of myself by the picture.*	(neg)
s4	*The awful soundtrack was disgusting and made me puke.*	(neg)
s5	*My house is green with a big yellow door.*	(neu)
s6	*The movie is there and I am here.*	(neu)
s7	*It is the first day of June.*	(neu)
s8	*Each glass bottle has been paid for each metal can.*	(neu)
s9	*A woman's hair is wonderful.*	(pos)
s10	*The mate's strength was amazing.*	(pos)
s11	*Ellie was an inspiration to her friends and family.*	(pos)
s12	*I was extremely happy with the movie.*	(pos)

Table 5.14: System preferences. ws: without sentiment, wcd: word context and tree distance, wl: word level

	ws	wcd	wl
mean	1.97	1.88	1.84
variance	0.59	0.68	0.86

Table 5.15: One- and two-tailed t-test results, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level, $\alpha = 0.05$

	one-tailed P	two-tailed P
ws/wcd	0.06	0.12
ws/wl	0.01	0.01
wl/wcd	0.28	0.55

sentiment with an average rank of 1.50

Table 5.18 shows preference results for users with different experience levels. Developer participants generally follow the tendency of the overall results, evaluating better the systems with sentiment than without. The P-values of the t-test for the developer participants are listed in Table 5.19 there are significant differences between both systems with sentiment and the system without sentiment, but no significant differences between the two systems with sentiment.

The user participants, on contrary, prefer the system without sentiment. However, the general tendency of the user participants is a rather good ranking of all systems, i.e. they considered more often that several systems were equally good. In any case, only two persons reported to be experienced user, with no further details how far this experience goes, which has no statistical importance, therefore no t-test is performed for the user participants.

The participants without experience preferred the system without sentiment and the system with *word level* sentiment, and pretty much discarded the system with *word context and tree distance*. The t-test results for the no-expert participants are listed in Table 5.20. The results show that there is a significant difference between the system without sentiment and the *word context and tree distance*, but no significant difference in other combinations. However, although the difference between the *word level* and the *word context and tree distance* is not significant, it is much bigger than the difference between the system without sentiment and the *word level* system. In general, and especially for participants without experience, the *word level* system has the highest variance.

5.5 Discussion

This chapter was dedicated to expressive speech synthesis with deep neural networks. For this, a DNN based speech synthesis system was trained on an audiobook, where additionally, sentiment input predicted by the Stanford sentiment

Table 5.16: System preferences for positive, negative and neutral sentences. ws: without sentiment, wcd: word context and tree distance, wl: word level

	ws	wcd	wl
positive mean	1.84	1.85	1.71
positive variance	0.54	0.76	0.54
negative mean	2.06	1.96	1.84
negative variance	0.52	0.67	1.1
neutral mean	2	1.83	1.96
neutral variance	0.71	0.6	0.95

Table 5.17: One- and two-tailed t-test results for positive, negative and neutral sentences, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level, $\alpha = 0.05$

	Neg:1-t.	Neg:2-t.	Neu:1-t.	Neu:2-t.	Pos:1-t.	Pos:2-t.
ws/wcd	0.12	0.24	0.01	0.02	0.46	0.92
ws/wl	0.01	0.02	0.36	0.72	0.04	0.08
wl/wcd	0.17	0.34	0.15	0.3	0.08	0.15

parser was added to train the system. Five different configurations were tested, among them including the sentiment probability on sentence level, word level, including word context and hierarchical tree distance, also with combination with sentence level, and using vector embeddings instead of probabilities.

A pitch analysis of two corpora was conducted, one neutral speech corpus, and the audiobook used in the perceptual experiments, in order to evaluate if there are differences in pitch depending on the sentence sentiment. Also, individual examples of pitch predictions with and without sentiment were visualized and studied. These results show that there is a clear influence on pitch by sentiment. However, the performance of the Stanford parser is not always optimal. Also, not always lies the sentence accent on the most positive or negative word, which sometimes yields a rare accentuation by the DNN.

Two perceptual experiments were conducted with test sentences synthesized using the different sentiment input configurations. The preliminary experiment aimed at preselecting the best systems, showing differences in performance for positive and negative sentences. The second experiment compared only two sentiment systems with the system without sentiment. The overall results yield that the systems with sentiment are better. Also, there are differences between positive, negative, and neutral sentences. However, when the results are separated by the experience of the participants with speech technology, there are important differences between the groups. The developer confirm and accentuate the overall results. The participants without any experience often preferred the system without sentiment features. Those with user experience had a different tendency, although there were only two of them, making the interpretation of their results statistically irrelevant. The best performing system, the word

Table 5.18: System preferences between developer participants, user participants, and participants without experience with speech technology. ws: without sentiment, wcd: word context and tree distance, wl: word level

	ws	wcd	wl
developer mean	2.01	1.79	1.77
developer variance	0.67	0.6	0.74
user mean	1.75	1.79	1.88
user variance	0.46	0.69	0.72
unexpert mean	1.94	2.08	1.96
unexpert variance	0.48	0.78	1.17

Table 5.19: One- and two-tailed t-test results for developer participants, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level, $\alpha = 0.05$

	one-tailed P	two-tailed P
ws/wcd	0.00	0.00
ws/wl	0.00	0.00
wl/wcd	0.22	0.44

Table 5.20: One- and two-tailed t-test results for no-expert participants, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level, $\alpha = 0.05$

	one-tailed P	two-tailed P
ws/wcd	0.02	0.03
ws/wl	0.44	0.87
wl/wcd	0.06	0.12

level sentiment system, has also the highest variance. This is probably due to the fact that this system yields the strongest and most varied accentuations since it is driven by word-level sentiment. This can be perceived sometimes as good and sometimes as bad.

The results obtained in the experiments show the general potential of neural network based synthesis in combination with expressive information derived from text. The results show the general preference for the best performing system using this information. However, it also shows that different designs of the input yield very different results in system performance, which probably means, that there is a lot more room for improvement.

Furthermore, the sentiment parser is trained on movie reviews, and the acoustic model on an audiobook. The consequence is that many sentences which are positive or negative in one domain, are different in the other domain. Also, movie reviews are usually written, and even if spoken, often with neutral voice.

This discrepancy probably lowers the quality of the prediction, of the training, and of the synthesis. On the other hand, the original audiobook by itself, is not very expressive.

Future work should aim, first, at improving these conditions, the database and the sentiment parser. After that, the way how the sentiment information is used in the system should be studied and improved. One of the main point regarding this is that there should be a connection between the sentiment (or other) sentence embeddings and the actual acoustics. A good investigation could be to train the sentiment analysis in such a way that the sentiment output is adjusted not only to the labels on text level, but also to the acoustics. Features like i-vectors or i-vector based combinations proposed in Chapter 3 could be used instead of labels automatizing the process. This technique could also work for other semantic embeddings adjusting them to the acoustics and improving them for the expressiveness.

Chapter 6

Discussion

Getting to the end of this work, a few last points are presented in this diskussion. Section 6.1 recalls the thesis goals and gives an overview of the whole work, and Section 6.2 drives final conclusions and speaks about future work. Finally, Section 6.3 shows a list of published contributions related to this thesis.

6.1 Summary

The subject of the present thesis is the expressive speech synthesis. The difference between expressive speech synthesis and "general" speech synthesis is that it can read text with different expressions. In many emotional or expressive speech synthesis systems this capability was already explored, creating emotional voices like "happy" or "sad", reading positive or negative news, etc. However, the focus of this thesis lies on unsupervised, automatic and labelless learning for expressive uncontrolled resources, like audiobooks, in order to gain training data; and to derive expressiveness from text, which can also be used to gain training data, or to automatically read expressive texts like books adapting the voice to the passages. Recalling the main hypothesis:

1. It is possible to define expressive voices from clusters of data in the acoustic domain, applying unsupervised methods to build the clusters, i.e. no labels of human interpretation are permitted to define the voices or the data in the clusters.
2. It is possible to improve the expressiveness of a synthetic voice using in the training process semantic features which codify some sort of expressive information and are obtained fully automatically.

and the main questions:

1. How is expressiveness represented in the acoustic and the semantic domains?
2. Is it possible to define reliable features for each of the domain which reflect the expressiveness?

3. If relevant features can be defined, how can they be used to prove the hypothesis in each case?

we can summerize the work as follows. First, the acoustic domain has been studied in order to find acoustic features which best represent expressiveness in speech. Second, semantic vector representations, especially derived from neural networks, have been used to predict expressiveness from text. On the other hand, two distinct paradigms of TTS systems have been explored for the task. The well established HMM-based synthesis, which provides enough flexibility to implement expressive voices, especially using speaker adaptive training. And the novel approach based on neural networks, where expressive speech is a less explored issue so far.

Regarding the acoustic domain, different features have been tested to represent expressive speech, in connection with speech synthesis. For this, different feature combinations have been extracted for the utterances of two corpora, an unsupervised clustering was performed on them, and the clusters were evaluated calculating their entropy. Then, HMM-based speech synthesis system was trained on the clusters using speaker adaptive training to create voices. These voices were then evaluated in a novel evaluation form where the participants were asked to edit an audiobook paragraph using the synthetic voices, showing clear preferences of certain voices for specific characters. The main novelty in this part is the usage of *i-vectors* to represent expressive speech in multi-speaker domains. I-vectors come from speaker recognition and turned out to be very suitable for the given task, outperforming other traditional features and sophisticated state-of-the-art feature sets used in emotional challenges and similar task. An interesting observation was made, that at least in the given task and for given evaluation technique, traditional features based on pitch outperformed the state-of-the-art feature sets, questioning, how well designed they actually are.

Regarding the semantic domain, numerical semantic representations in vector space have been used to represent the text and automatically derive expressiveness from it. One of the most important goals was to avoid labeling. First, labeling is very costly in terms of manual work. Second, often it is not really clear how to define labels in order to account for all the possible expressions, emotions, speaking styles, etc., especially for open multi-speaker domains.

A preliminary experiment with vector representations was performed, visualizing that, at least to some extent, expressiveness is encoded in text and could be predicted. Then, more sophisticated embeddings, derived from neural networks, have been used to perform a prediction from semantics to acoustics, also here yielding results which show that at least some information can be derived. Two perceptual evaluations were designed to test the idea. First, for each sentence of two book paragraph, acoustic feature vector were predicted from semantic representations and used as cluster centroids in an acoustic feature space, selecting a nearest-neighbor cluster to train an HMM-based TTS via speaker adaptive training, creating an individual voice for each of the paragraphs. The test results showed clear preference of the expressive reading in comparison to the reading with the neutral voice. In the second task, for emotional key words, acoustic feature vectors were predicted from the semantics, and again, these were used as cluster centroids to select data for speaker adaptation. This approach is ba-

sically a search engine for expressive/emotional speech based on semantics. The emotional voices were successfully evaluated in a preference test.

Following the newly emerged paradigm of neural network based speech synthesis, a DNN-based speech synthesis system was implemented using sentiment information derived from text. The sentiment information is predicted by the Stanford sentiment parser, where probabilities and semantic vector embeddings are calculated for each unit in the input text. Two corpora were studied for the effects of sentiment on pitch, showing that there are changes in pitch depending on the sentiment value of the sentences, especially for the audiobook corpus, which is an expressive speech corpus. Individual examples were studied showing how sentiment influences the DNN output for pitch, changing accents and accentuation strength in comparison to the model without sentiment. Further, two perceptual experiments have been conducted using the samples synthesized by the DNN-based TTS using different configurations. The first experiment aimed at choosing the best system configurations, revealing that the system performance is not equal for positive and negative sentences. The second experiment involved more participants, especially those who were not experienced with speech technology, showing important differences in preferences of experts and non-experts in speech technology. The general results show that the sentiment information can be very useful to improve the synthesis, but it needs to be carefully designed. Also, some discrepancies regarding the sentiment prediction itself should be remodeled, like the fact that it is trained using movie reviews and the acoustic models are trained using an audiobook. These different domains yield important semantic differences for the sentiment prediction.

Regarding the hypothesis we can confirm both of them. It is possible to define expressive voices from clusters of data in the acoustic domain, applying unsupervised methods to build the clusters, i.e. no labels of human interpretation are permitted to define the voices or the data in the clusters. Also, it is possible to improve the expressiveness of a synthetic voice using in the training process semantic features which codify some sort of expressive information and are obtained fully automatically. In both cases though, there are limitations, at least encountered in the focus of the presented work. In order to overcome these limitations it is necessary to substantiate and redefine several aspects, especially regarding new ways made possible by newly emerged technologies. The next section amplifies the conclusions and the future work.

6.2 Conclusions and future work

As seen for the given tasks, progress has been made in order to achieve the proposed goals, and also new issues have been opened. For the acoustic part, i-vectors have proved to be the most suitable feature for the given task, outperforming other state-of-the-art and traditional features. However, this is a generalization, where in individual cases things might look differently. For instance, for the mono speaker databases, the feature issue is not really resolved, and, at least in the present studies, the most simple and traditional features could not be outperformed. Also, the speaker specific differences, where among different speakers different features yield different performance. It is difficult to determine when which feature will perform better, which is also consistent

with other studies where opinions about the best features diverge. What can be concluded is, that i-vectors did perform best in the multi-speaker domain, and that maybe, the best way to continue research in the acoustic domain is to leave acoustic low-level features and concentrate on techniques which try to separate information codified in the low-level features, similar to i-vectors.

For the part of the semantic domain, also here, it has become clear, that expressiveness is encoded somehow in the semantics, and that with modern techniques a semantic-acoustic mapping is possible. However, there is a lot of space of improvement. First, there is a lack of a systematic formulation, of how expressiveness can be derived from text, which is not trivial since the exact relation between the semantics and the acoustics is not totally clear, except for key words. Models like deep neural networks have shown to learn from semantics in order to predict acoustics in the sense of expressiveness, but the relations they learn still seem to be rather superficial and sometimes not intuitive. On the other hand, the role of the context and world knowledge is not totally clear. First, intuitively it seems very important for predicting expressiveness. However, where some experiments showed that it was crucial for improving the prediction, others showed that there was no improvement. Also here, a possible conclusion is that, a more systematic formulation of this knowledge is needed, on the other hand, a well designed approach of how to introduce it to the speech synthesis system.

Also the semantic representations per se should be reconsidered and remodeled for a better representation of expressiveness. For instance, the sentiment embeddings have high potential, but need to be trained on corresponding domains in order to make accurate predictions. Possibly, being sentiment one of the emotional dimensions formulated by Kehrein (2002), other dimensions of his model could be derived from text. Another viable option, as described in Section 5.5, is to train sentiment (or other) embeddings using acoustic criterion of underlying speech, such that the embeddings not only reflect whatever is interpretable from text, but also the actual acoustics.

A final comment about the new trends in speech synthesis in general: When this thesis started, the focus lied on statistical training methods, however, mean awhile a real "boom" of neural network based synthesis changed the synthesis paradigm in general. While it is true that many findings of this thesis are more general and rather independent of the underlying system architecture, the way how this knowledge is implemented in the system changes radically though, when applied to NN-based synthesis. This in turn, allows for and requires to reconsider the architectures and th system design, at least when dealing with expressive speech and opens many new ways and possibilities for future work. Apart of the joined training of vector embeddings regarding acoustics, mentioned above, possible approaches could for instance exploit large(r) data bases; different types of architectures, like for example parallel architectures which perform different tasks at once; transfer learning; and a long etc.

In German, there is a idiom, "Eierlegende Wollmilchsau", which translated means something like "oviparous wool-milk-pig". It describes a farm animal which has only advantages, i.e. lays eggs, produces wool and milk, and also can be eaten. This figure is a metaphor for something which is fully advantageous and accounts for all the needs. The last conclusion of the thesis is that there is

no "oviparous wool-milk-pig" in expressive speech synthesis so far and there is still a long way to go to breed one.

6.3 Published contributions

I. Jauk and A. Bonafonte. *Direct expressive voice training based on semantic selection.* In Proceedings of Interspeech, pages 3181–3185, San Francisco (USA), 2016.

I. Jauk and A. Bonafonte. *Prosodic and spectral ivectors for expressive speech synthesis.* In Proceedings of Speech Synthesis Workshop 9, pages 59–63, 2016.

I. Jauk, A. Bonafonte, P. López-Otero, and L. Docio-Fernandez. *Creating expressive synthetic voices by unsupervised clustering of audiobooks.* In Proceedings of Interspeech, pages 3380–3384, Dresden (Germany), 2015.

I. Jauk, A. Bonafonte, and S. Pascual. *Acoustic feature prediction from semantic features for expressive speech using deep neural networks.* In Proceedings of EUSIPCO, pages 2320–2324, Budapest (Hungary), 2016.

Bibliography

S. Achanta, T. Godambe, and S.V. Gangashetty. An investigation of recurrent neural network architectures for statistical parametric speech synthesis. *Proceedings of Interspeech*, pages 859–863, 2015.

J. Adell, D. Escudero, and A. Bonafonte. Production of filled pauses in concatenative speech synthesis based on the underlying fluent sentence. *Speech Communication*, 54:459–776, 2012.

P.D. Agüero and A. Bonafonte. Phrase break prediction: a comparative study. In *XIX Congreso de la Sociedad Española para el procesamiento del Lenguaje Natural*, 2003.

P.D. Agüero and A. Bonafonte. Intonation modeling for tts using a joint extraction and prediction approach. In *Proceedings of Speech Synthesis Workshop 6 (SSW6)*, pages 67–72, 2004a.

P.D. Agüero and A. Bonafonte. Phrase break prediction using a finite state transducer. In *Proceedings of Advanced Speech Technologies*, 2004b.

F. Alías Pujol. *Conversión de texto en habla multidominio basada en selección de unidades con ajuste subjetivo de pesos y marcado robusto de pitch*. PhD thesis, Universitat Ramon Llull, 2006.

T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proceedings of ICSLP*, pages 1137–1140, 1996.

S.O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi. Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017. URL http://arxiv.org/abs/1702.07825.

M. Arnela, S. Dabbaghchian, R. Blandin, O. Guasch, O. Engwall, A. Van Hirtum, and X. Pelorson. Influence of vocal tract geometry simplifications on the numerical simulation of vowel sounds. *The Journal of the Acoustic Society of America*, 140(3):1707–1718, 2016.

J. Baker. The dragon system–an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):24–29, 1975.

R. Barra-Chicote, J. Yamagishi, S. King, J. Montero, and J. Macias-Guarasa. Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. *Speech Communication*, 52:394–404, 2010.

J.R. Bellegarda. Unsupervised document clustering using latent semantic density analysis, January 12 2012. URL https://www.google.ch/patents/US20120011124. US Patent App. 12/831,909.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic model. *Journal of Machine Learning Research*, (3):1137–1155, 2003.

P. Birkholz. *3D-Artikulatorische Sprachsynthese*. PhD thesis, Universität Rostock, 2005.

P. Birkholz, B.J. Kröger, and C. Neuschaefer-Rube. Articulatory synthesis of words in six voice qualities using a modified two-mass model of the vocal folds. In *Proceedings of the First International Wokrshop on Performative Speech and Singing Synthesis*, 2011.

P. Birkholz, L. Martin, K. Willmes, and B.J. Kröger. The contribution of phonation type to the perception of vocal emotions in german: an articulatory synthesis study. *Journal of the Acoustic Society of Amercia*, 137(3):1503–1512, 2015.

P. Boersma and D Weenink. Praat: doing phonetics by computer (version 5.4.07), 2015. URL http://www.praat.org/.

T. Bonafonte, P. Aguero, J. Adell, J. Perez, and A. Moreno. Ogmios: the UPC text-to-speech synthesis system for spoken translation. In *Proceedings of TC-STAR Workshop on Speech-to-Speech Translation*, pages 199–204, 2006.

S. Breuer. *Multifunktionale und multilinguale Unit-Selection-Sprachsynthese*. PhD thesis, Universität Bonn, 2009.

S. Breuer and W. Hess. The Bonn Open Synthesis System 3. *International Journal of Speech Technology*, 13(2):75–84, 2010.

F. Burckhardt and W.F. Sendelmeier. Verification of acoustical correlates of emotional speech using formant synthesis. In *Proceedings of ISCA Workshop on Speech and Emotion*, pages 151–156, 2000.

J.E. Cahn. Generation of affect in synthesized speech. In *Proceedings of American Voice I/O Society*, pages 251–256, 1989.

N. Campbell. Syllable-based segmental duration. *Talking Machines: Theories, Models and Designs*, pages 211–224, 1992.

N. Campbell. Conversational speech synthesis and the need for some laughter. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):1171–1179, 2006.

J.P. Carbal and L.C. Oliveira. Emo voice: a system to generate emotions in speech. In *Proceedings of Interpseech*, pages 1798–1801, 2006.

A. Chalamandaris, P. Tsiakoulis, S. Karabetsos, and S. Raptis. Using audio books for training a text-to-speech system. In *Proceedings of LREC*, pages 3076–3080, 2006.

A. Chalamandaris, P. Tsiakoulis, S. Karabetsos, and S. Raptis. The ilsp/innoetics text-to-speech system for the blizzard challenge 2014. In *Proceedings of Blizzard Challenge*, 2014.

Blizzard Challenge. URL http://www.festvox.org/index.html.

F. Charpentier and M. Stella. Diphone synthesis using an overlpa-add technique for speech waveforms concatanation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2015–2018, 1986.

L. Chen, M.J.F. Gales, N. Braunschweiler, M. Akamine, and K. Knill. Integrated expression prediction and speech synthesis from text. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):323–335, 2014.

L.-H. Chen, Y. Jiang, M. Zhou, Z.-H. Ling, and L.-R. Dai. The ustc system for blizzard challenge. In *Proceedings of Blizzard Challenge*, 2016.

S.-H. Chen, Sh.-H. Hwang, and Y.-R. Wang. An rnn-based prosodic information synthesizer for mandarin text-to-speech. *IEEE Transactions on Speech and Audio Processing*, pages 226–239, 1998.

F.S. Cooper, A.M. Liberman, and J.M. Borst. The interconversion of audible and visible patterns as a basis for research in the perception of speech. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 318–325, 1951.

R. Córdoba, J.A. Vallejo, J.M. Montero, J. Gutierrez-Arriola, M.A. López, and J.M. Pardo. Automatic modeling of duration in a spanish text-to-speech system using neural networks. In *Proceedings of Eurospeech*, 1999.

R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, and J.G. Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, pages 32–80, 2001.

N. Dehak. *Discriminative and Generative Approaches for Long- and Short-term Speaker Characteristics Modeling: Application to Speaker Verification*. PhD thesis, 2009. AAINR50490.

N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788–798, 2011.

P. Delattre. From acoustic cues to distinctive features. *Phonetica*, pages 198–230, 1968.

Romaji Desu. URL http://www.romajidesu.com/kanji/.

M.M. Deza and E. Deza. *Encyclopedia of distances*. Springer, 2009.

Cambridge dictionary, a. URL http://dictionary.cambridge.org/.

Oxford dictionary, b. URL https://www.oxforddictionaries.com/.

J. Duchi and Y. Hazan, E. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

H. Dudley. The vocoder. *Bell Labs Rec.*, (17):122–126, 1939.

H.K. Dunn. The calculation of vowel resonances, and electrical vocal tract. *Journal of the Acoustic Society of America*, (22):740–753, 1950.

P. Ekman. Universals and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation*, 19:207–282, 1972.

D. Erro, I. Sainz, E. Navas, and I. Hernaez. Improved HNM-based vocoder for statistical synthesizers. In *Proceedings of Interspeech*, pages 1809–1812, 2011a.

D. Erro, I. Sainz, E. Navas, and I. Hernaez. HNM-based mfcc+f0 extractor applied to statistical speech synthesis. In *Proceedings of ICASSP*, pages 4728–4731, 2011b.

D. Erro, I. Hernaez, E. Navas, A. Alonso, H. Arzelus, I. Jauk, N.Q. Hy, C. Magariños, R. Perez-Ramon, M. Sulir, X. Tian, X. Wang, and J. Ye. Zuretts: Online platform for obtaining personalized synthetic voices. In *Proceedings of eNTERFACE*, pages 17–25, Bilbao (Spain), 2014.

D. Erro, I. Hernaez, A. Alonso, D. Garcia-Lorenzo, E. Navas, J. Ye, H. Arzelus, I. Jauk, N.Q. Hy, C. Magariños, R. Perez-Ramon, M. Sulir, X. Tian, and X. Wang. Personalized synthetic voices for speaking impaired: Website and app. In *Proceedings of Interspeech*, pages 1251–1254, Dresden (Germany), 2015.

D. Escudero, V. Cardeñoso, and A. Bonafonte. Corpus based extraction of quantitative prosodic parameters of stress groups in spanish. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference*, volume 1, pages I–481–I–484, 2002.

Erro D. et al. Zure tts. URL http://aholab.ehu.es/zuretts/.

F. Eyben. The opensmile book, 2016. URL http://opensmile.audeering.com/.

F. Eyben, S. Buchholz, N. Braunschweiler, J. Latorre, V. Wan, M. Gales, and K. Knill. Unsupervised clustering of emotion and voice styles for expressive TTS. In *Proceedings of ICASSP*, pages 4009–4012, 2012.

F. Eyben, F. Weninger, F. Gross, and B. Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. pages 835–838, October 2013.

J.W.A. Fackrell, H. Vereecken, J.P. Martens, and B.V. Coile. Multilingual prosody modelling using cascades of regression trees and neural networks. In *Proceedings of Eurospeech*, 1999.

G. Fant. Speech communication research. *Ing. Vetenskaps of Royal Swedish Academy of Engineering Sciences*, 24:331–337, 1953.

G. Fant. *Acoustic Theory of Speech Production*. Mouton, 2 edition, 1970.

M. Farrús, J. Hernando, and P. Ejarque. Jitter and shimmer measurements for speaker recognition. In *8th Annual Conference of the International Speech Communication Association*, pages 778–781, 2007.

A. Febrer, J. Padrell, and A. Bonafonte. Modeling phone duration: application to catalan tts. In *Proceedings of Speech Synthesis Workshop 3 (SSW3)*, pages 43–46, 1998.

R. Fernandez, B. Rendel, A. Ramabhadran, and R. Hoory. Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks. *Proceedings of Interspeech*, pages 2268–2272, 2014.

H. Fujisaki and H. Kawai. Modeling the dynamic characteristics of voice fundamental frequency with applications to analysis and synthesis of intonation. In *Working Group on Intonation, 13th International Congress of Linguists*, pages 57–70, 1982.

N. Geschwind. Language and the brain. *Scientific American*, 226:76–83, 1972.

M Giustiniani and P. Pierucci. Phonetic ergodic hmm for speech synthesis. In *Proceedings of EUROSPEECH*, pages 349–352, 1991.

GNU. GSL - GNU scientific library. URL http://www.gnu.org/software/gsl/.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

D. Govind and S.R. Mahadeva Prasanna. Expressive speech synthesis: a review. *International Journal of Speech Technology*, 16(2):237–260, 2013.

R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

W. Hamza, R. Bakis, E. Eide, M. Picheny, and J. Pitrelli. The IBM expressive speech synthesis system. In *Proceedings of ICSLP*, pages 2577–2580, 2004.

J. Hart and A. Cohen. Intonation by rule: A perceptual quest. *Journal of Phonetics*, 1:309–327, 1973.

F. Hinterleitner, G. Neitzel, S. Möller, and Ch. Norrenbrock. An evaluation protocol for the subjective assessment of text-to-speech in audiobook reading tests. In *Proceedings of Blizzard Challenge*, 2011.

V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte, and A. Nogueiras. Interface databases: Design and collection of a multilingual emotional speech database. In *LREC*, pages 2024–2028, 2002.

Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia. Fusion of multiple parameterisations for dnn-based sinusoidal speech synthesis with multi-task learning. *Proceedings of Interspeech*, pages 854–858, 2015.

A. Iida, N. Campbell, S. Iga, G. Higuchi, and M. Yasumura. A speech synthesis system for assisting communications. In *Proceedings of ISCA Workshop on Speech and Emotion*, pages 167–172, 2000.

S. Imai. Cepstral analysis synthesis on the mel frequency scale. In *Acoustics, Speech, and Signal Processing (ICASSP)*, pages 93–96, 1983.

International Phonetic Association (IPA). *IPA Handbook*. Cambridge University Press, 8 edition, 2007.

I. Jauk. Rhythmusmodellierung im sprechgesang, 2010.

I. Jauk and A. Bonafonte. Direct expressive voice training based on semantic selection. In *Proceedings of Interspeech*, pages 3181–3185, 2016a.

I. Jauk and A. Bonafonte. Prosodic and spectral ivectors for expressive speech synthesis. In *Proceedings of Speech Synthesis Workshop 9*, pages 59–63, 2016b.

I. Jauk, A. Bonafonte, P. López-Otero, and L. Docio-Fernandez. Creating expressive synthetic voices by unsupervised clustering of audiobooks. In *Interspeech 2015*, pages 3380–3384, 2015.

I. Jauk, A. Bonafonte, and S. Pascual. Acoustic feature prediction from semantic features for expressive speech using deep neural networks. In *Proceedings of EUSIPCO*, pages 2320–2324, 2016.

B.-H. Juang. On the hidden markov model and dynamic time warping for speech recognition - a unified view. *AT and T Technical Journal*, 63(7):1213–1243, 1984.

H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(3-4):187–207, 1999.

R. Kehrein. The prosody of authentic emotions. In *Proceedings of Speech Prosody*, pages 423–426, 2002.

P. Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *Technical report CRIM-06/08-13 Montreal*, 2005. URL http://www.crim.ca/perso/patrick.kenny/.

P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice modeling with sparse training data. *IEEE Transactions on Speech and Audio Processing*, 13(3): 345–354, 2005.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

R. Klabunde, K.U. Carstensen, C. Ebert, C. Endriss, S. Jekat, and H. Langer. *Computerlinguistik und Sprachtechnologie*. 2004.

D. Klatt. Dennis klatt's 'history of speech synthesis' archive. URL http://www.cs.indiana.edu/rhythmsp/ASA/Contents.html.

D. Klatt. Review of text-to-speech conversion for english. *Journal of the Acoustic Society of America*, (82):737–793, 1987.

D.H. Klatt. Acoustic theory of terminal analog speech synthesis. In *Acoustics, Speech, and Signal Processing (ICASSP)*, pages 131–135, 1972.

D.H. Klatt. Interaction between two factors that influence vowel duration. *Journal of the Acoustic Society of America*, 5:1102–1104, 1973.

D. Klein and C.D. Manning. Accurate unlexicalized parsing. *ACL*, pages 423–430, 2003.

K.J. Kohler. A model of german intonation. In *Studies in German Intonation*, pages 295–368. Institut für Phonetik und digitale Sprachverarbeitung der Universität Kiel, 1991.

B.J. Kröger and P. Birkholz. Articulatory synthesis of speech and singing: State of the art and suggestions for future research. *Multimodal Signals: Cognitive and Algorithmic Issues*, pages 306–319, 2009.

K. Kuttler. *An introduction to linear algebra*. 2007. URL http://www.math.byu.edu/~klkuttle/Linearalgebra.pdf.

G.N. Lance and W.T. Williams. Mixed-data classifictory programs i - agglomerative systems. *Australian Computer Journal*, pages 15–20, 1967.

W. Lawrence. The synthesis of speech from signals which have a low information rate. *Communication Theory*, pages 460–469, 1953.

M. Liberman. *The Intonational System of English*. PhD thesis, MIT, 1975.

A. Ljolje and F. Fallside. Synthesis of natural sounding pitch contours in isolated utterances using hidden markov models. *IEEE Transactions on Audio, Speech and Language Processing*, ASSP-34:1074–1080, 1986.

P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo. iVectors for continuous emotion recognition. In *Proceedings of Iberspeech 2014*, pages 31–40, 2014.

J. Lorenzo Trueba. *Design and Evaluation of Statistical Parametric Techniques in Expressive Text-To-Speech: Emotion and Speaking Styles Transplantation*. PhD thesis, E.T.S.I. Telecomunicación (UPM), 2016.

J. Lorenzo-Trueba, R. Barra-Chicote, J. Yamagishi, O. Watts, and J.M. Montero. Towards speaking style transplantation in speech synthesis. In *Proceedings of 8th ISCA Speech Synthesis Workshop*, pages 159–163, 2013.

J. Lorenzo-Trueba, R. Barra-Chicote, J. Yamagishi, and J.M. Montero. Towards cross-lingual emotion transplantation. In *Proceedings of Iberspeech 2014*, pages 199–208, 2014.

H. Lu, S. King, and O. Watts. Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. *Proceedings of Speech Synthesis Workshop (SSW8)*, pages 281–285, 2013.

T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis from hmms using dynamic features. In *Acoustics, Speech, and Signal Processing (ICASSP)*, pages 389–392, 1996.

P. Menzerath and A. de Lacerda. *Koartikulation und Steuerung*. Dümmler (Bonn), 1933.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. *ArXiv e-prints*, October 2013.

J.M. Montero, J. Gutierrez-Arriola, J. Colas, E. Enriquez, and J.M. Pardo. Analysis and modeling of emotional speech in spanish. In *Proceedings of ICPhS*, pages 671–674, 1999.

I.R. Murray and J.L. Arnott. Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion. *The Journal of the Acoustic Society of America*, pages 1097–1108, 1993.

I.R. Murray and J.L. Arnott. Implementation and testing of a system for producing emotion by rule in synthetic speech. *Speech Communication*, 16:369–390, 1995.

M.A. Musen. The protégé project: A look back and a look forward. *AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence*, 1(4), 2015. DOI: 10.1145/2557001.25757003.

B. Möbius. *Ein quantitatives Modell der deutschen Intonation. Analyse und Synthese von Grundfrequenzverläufen.* Niemeyer, Tübingen, 1993.

Y. Nesterov. A method for uncontrained convex minimization problem with the rate of convergence o(1/k2). *Doklady ANSSSR (translated as Soviet.Math.Docl)*, 269:543–547, 1983.

T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi. A style control technique for HMM-based expressive speech synthesis. *IEICE Transactions on Information and Systems*, (9):1406–1413, 2005.

T. Nose, M. Tachibana, and T. Kobayashi. Hmm-based style control for expressive speech synthesis with arbitrary speaker's voice using model adaptation. *IEICE Transactions on Information and Systems*, E92-D(3):489–497, 2009.

A. Ortega and A. Miguel. Adaptation and normalization techniques (especially) for automatic speech recognition. *Applications of speech technologies*, pages 16–71, 2014.

B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *ACL*, pages 115–124, 2005.

S. Pascual and A. Bonafonte. Multi-ouput rnn-lstm for multiple speaker speech synthesis with α-interpolation model. *Proceedings of Speech Synthesis Workshop (SSW9)*, pages 119–124, 2016a.

S. Pascual and A. Bonafonte. Prosodic break prediction with rnns. In *Proceedings of Iberspeech*, pages 64–72, 2016b.

J.W. Pennebaker. *The Secret Life of Pronouns.* 2011.

J. Pennington, R. Socher, and C.D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

J.B. Pierrehumbert. *The Phonology and Phonetics of English Intonation*. PhD thesis, MIT, 1980.

T.V. Polyàkova. *Grapheme-to-phoneme conversion in the era of globalization*. PhD thesis, Universitat Politècnica de Catalunya, 2015.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

Princeton. About wordnet, 2010. URL http://wordnet.princeton.edu.

J. Pérez and A. Bonafonte. Automatic voice-source parameterization of natural speech. In *Proceedings of 9th European Conference on Speech Communication and Technology Interspeech 2005*, pages 1065–1068, Lisboa, Portugal, September 2005.

N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145 – 151, 1999. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(98)00116-6. URL http://www.sciencedirect.com/science/article/pii/S0893608098001166.

Y. Qian, Y. Fan, W. Hu, and F.K. Soong. On the training aspects of deep neural networks (dnn) for parametric tts synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3829–3833, 2014.

L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. PTR Prentice-Hall, Inc., 1993.

S.K. Rallabandi, A. Vadapalli, S. Achanta, and S.V. Gangashetty. Iiit hyderbad's submission to the blizzard challenge 2015. In *Proceedings of Blizzard Challenge*, 2015.

S. Reese, G. Boleda, L. Cuadros, M. Padró, and G. Rigau. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC10)*, pages 1418–1421, 2010.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

K. Richmond, R.A. Clark, and S. Fitt. Robust lts rules with the combilex speech technology lexicon. *Proceedings of Interspeech*, pages 1295–1298, 2009.

M. Riley. Tree-based modeling in segmental duration. *Talking Machines: Theories, Models and Designs*, pages 265–273, 1992.

M. Rozak. Text-to-speech designed for a massively multiplayer online role-playing game (mmorpg). In *Proceedings of Blizzard Challenge*, 2007.

K. Saino, H. Zen, Y. Hankaku, A. Lee, and K. Tokuda. Hmm-based singing voice synthesis system. In *Proceedings of Interspeech*, pages 1141–1144, 2006.

Expressive Synthetic Speech Samples. URL http://emosamples.syntheticspeech.de/.

J. Sato and S. Morishima. Emotion modeling in speech production using emotion space. *IEEE International Workshop on Robot and Human Communication*, pages 1171–1179, 1996.

U. Schade. Konnektionistische sprachproduktion. In *Psycholinguistische Studien*, pages 183–196, 1999.

H. Schlossberg. Three dimensions of emotions. *Psychological Review*, 61(2): 81–88, 1954.

M. Schröder. Expressive speech synthesis: past, present, and possible futures. In *Affective Information Processing*, chapter 7, pages 111–126. 2009.

M. Schröder. Emotional speech synthesis - a review. In *Proceedings of EUROSPEECH*, pages 561–564, 2001.

B. Schuller. Recognizing affect from linguistic information in 3d continuous space. *IEEE Transactions on affective computing*, 2(4):192–205, 2000.

B. Schuller, R. Müller, M. Lang, and G. Rigoll. Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. In *Proceedings of Interspeech*, pages 805–808, 2005.

B. Schuller, A. Steidl, and A. Batliner. The interspeech 2009 emotion challenge. In *Proceedings of Interspeech*, pages 312–315, 2009.

B. Schuller, A. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and Sh. Narayanan. The interspeech 2010 paralinguistic challenge. In *Proceedings of Interspeech*, pages 2794–2797, 2010.

C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

J. Sotscheck, W. Endres, W. Hess, R. Hoffmann, M. Krause, A. Lacroix, H. Mangold, E. Paulus, and Wolf H.E. Itg empfehlung 4.3.1-01 terminologie der sprachakustik. *The Journal of the Acoustical Society of America*, 1996.

R. Sproat, A.W. Black, S. Chen, Sh. Kumar, M. Ostendorf, and Ch. Richards. Normalization of non-standard words. *Computer Speech and Language*, 15(3): 287–333, 2001.

SPTK. Speech signal processing toolkit ver. 3.6. URL http://sp-tk.sourceforge.net/.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

K.N. Stevens, S. Kasowski, and G. Fant. An electrical analog of the vocal tract. *Journal of the Acoustic Society of America*, (25):734–742, 1953.

S.S. Stevens, J. Volkman, and E.B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, January 1937.

J.Q. Stewart. An electrical analogue of the vocal organs. *Nature*, 110:311–312, 1922.

B. Sudhakar and R. Bensraj. An efficient sentence-based sentiment analysis for expressive text-to-speech using fuzzy neural network. *Research Journal of Applied Sciences, Engineering and Technology*, 8(3):378–386, 2014.

E. Szekely, J. Cabral, P. Cahill, and J. Carson-Berndsen. Clustering expressive speech styles in audiobooks using glottal source parameters. In *Proceedings of Interspeech*, pages 2409–2412, 2011.

E. Szekely, T.G. Csapó, B. Tóth, P. Mihajlik, and J. Carson-Berndsen. Synthesizing expressive speech from amateur audiobook recordings. In *Proceedings of Spoken Language Technology Workshop (SLT)*, pages 2409–2412, 2012.

M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi. Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing. *IEICE Transactions on Information and Systems*, E88-D(11): 2484–2491, 2005.

M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi. A style adaptation technique for speech synthesis using hsmm and suprasegmental features. *IEICE Transactions on Information and Systems*, E89-D(3):1092–1009, 2006.

S. Takaki and J. Yamagishi. Constructing a deep neural network based spectral model for statistical speech synthesis. *Recent Advances in Nonlinear Speech Processing*, 48:117–125, 2016.

T. Tan, Y. Qian, and K. Yu. Cluster adaptive training for deep neural network based acoustic model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24/3:459–468, 2016.

P. Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

P.A. Taylor. *A Phonetic Model of English Intonation*. PhD thesis, University of Edinburgh, 1992.

T. Tielemann and G. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning. *Technical Report*, 2012.

K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, A.W. Black, and T. Nose. The HMM-based speech synthesis system (HTS), 2008. URL http://hts.ics.nitech.ac.jp.

A. Trilla and F. Alias. Sentence based sentiment analysis for expressive text-to-speech. *IEEE Transactions on Audio, Speech and Language Processing*, 21(2):223–233, 2013.

G. Ungeheuer. *Elemente einer akustischen Theorie der Vokalartikulation*. Springer, 1962.

Macquarie University. A brief historical introduction to speech synthesis: A macquarie perspective. URL http://clas.mq.edu.au/speech/synthesis/history_synthesis/.

C. Valentini-Botinhao, Z. Wu, and S. King. Towards minimum perceptual error training for dnn-based speech synthesis. *Proceedings of Interspeech*, pages 869–873, 2015.

J. Van den Berg. Myoelastic-aerodynamic theory of voice production. *Journal of Speech, Language, anf Hearing Research*, 1:227–244, 1958.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio, 2016. URL https://deepmind.com/blog/wavenet-generative-model-raw-audio/.

J. van Santen. Assignment of segmental duration in text-to-speech synthesis. *Computer Speech and Language*, 8:95–128, 1994.

E. Vanmassenhove, J. Cabral, and F. Haider. Prediction of emotions from text using sentiment analysis for expressive speech synthesis. *Proceedings of Speech Synthesis Workshop (SSW9)*, pages 119–124, 2016.

P. Vary, U. Heute, and W. Hess. *Digitale Sprachsignalverarbeitung*. B. G. Teubner Stuttgart, 1998.

J. Vroomen, R. Collier, and S.J.L. Mozziconacci. Duration an intonation in emotional speech. In *Proceedings of EUROSPEECH*, pages 577–580, 1993.

W3C. Emotion markup language (emotionml), a. URL https://www.w3.org/TR/emotionml/.

W3C. Speech synthesis markup language (ssml), b. URL https://www.w3.org/TR/speech-synthesis/.

P. Wagner. *The rhythm of language and speech: constraining factors, models, metrics and applications*. 2008. URL http://www.uni-bielefeld.de/lili/personen/pwagner/pubs.html.

P. Wang, Y. Qian, F.K. Soong, L. He, and H. Zhao. Word embedding for recurrent neural network based tts synthesis. In *Proceedings of International conference on acoustics, speech and signal processing (ICASSP)*, pages 4879–4883, 2015.

X. Wang, S. Takaki, and J. Yamagishi. Investigating of using continuous representation of various linguistic units in neural network based text-to-speech synthesis. *IEICE Transactions on Information and Systems*, E99-D(10):2471–2480, 2016.

Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q.V. Le, Y. Agiomyrgiannakis, R. Clark, and R.A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017. URL http://arxiv.org/abs/1703.10135.

O. Watts. *Unsupervised Learning for Text-to-Speech Synthesis*. PhD thesis, University of Edinburgh, 2012.

J.C. Wells. Sampa computer readable phonetic alphabet. *Handbook of Standards and Ressources for Spoken Language Systems*, 4 (B), 1997.

R. Wiese. *Silbische und lexikalische Phonologie: Studien zum Chinesichen und Deutschen*. De Gruyter, 1988.

C. Wollermann. *Prosodie, nonverbale Signale, Unsicherheit und Kontext - Studien zur pragmatischen Fokusinterpretation*. PhD thesis, Universität Duisburg-Essen, 2012.

word2vec Tool for computing continuous distributed representations of words. URL http://www.gnu.org/software/gsl/.

Z. Wu and S. King. Investigating gated recurrent networks for speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5140–5144, 2016.

Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King. A study of speaker adaptation for dnn-based speech synthesis. *Proceedings of Interspeech*, 2015a.

Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4460–4464, 2015b.

J. Yamagishi. *Average-Voice-Based Speech Synthesis*. PhD thesis, Tokyo Institute of Technology, 2012.

J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Modeling of various speaking styles and emotions for hmm-based speech synthesis. In *Proceedings of Eurospeech*, pages 2461–2464, 2003.

J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Acoustic modeling of speaking styles and emotional expressions in hmm-based speech synthesis. *IEICE Transactions on Information and Systems*, E88-D(3):502–509, 2005.

M.D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL http://arxiv.org/abs/1212.5701.

H. Zen and H. Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4470–4474, 2015.

H. Zen and A. Senior. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3844–3848, 2014.

H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7962–7966, 2013.

A.W. Zewoudie, J. Luque, and J. Hernando. Jitter and shimmer measurements for speaker diarization. In *Proceedings of Iberspeech*, pages 21–30, 2014.

Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.