Dissertation ETH No. 21991

# A Data-driven Model for the Generation of Prosody from Syntactic Sentence Structures

A dissertation submitted to the
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by
SARAH HOFFMANN
Dipl. Inf.
born December 18, 1978
citizen of Germany

accepted on the recommendation of
Prof. Dr. Lothar Thiele, examiner
Prof. Dr. Bernd Möbius, co-examiner
Dr. Beat Pfister, co-examiner

2014

# Acknowledgements

I would like to thank Prof. Dr. Lothar Thiele and Dr. Beat Pfister for supervising my research and Prof. Dr. Bernard Möbius for taking part in the examination committee.

The former and current members of the ETH speech group helped a lot in getting me started in signal and speech processing, which were completely new fields to me. In particular, I would like to thank Tobias Kaufmann for the inspiring discussions on natural language processing and for writing what became the basis of the text parser for this thesis. Further, I want to especially thank Tofigh Naghibi for always being ready to listen to and give advice on pretty much any topic, research-related or otherwise.

I am indebted to the members of the OSM Stammtisch Zurich for providing ample and fun distraction from the hard research life and for politely ignoring the mood swings that came with the final year of writing down these pages. I would also like to thank my parents for their steady and unconditional support.

Finally, I would like to thank Martin Hoffmann for the thorough proofreading of the final version of this thesis. Much appreciated. Any remaining errors are entirely mine.

# Contents

# List of Abbreviations

| | |
|---|---|
| ANN | artificial neural network |
| ASR | automatic speech recognition |
| CART | classification and regression tree |
| DCG | direct clause grammar |
| $F_0$ | fundamental frequency |
| GMM | Gaussian mixture model |
| HMM | hidden Markov model |
| LPC | linear predictive coding |
| MFCC | Mel-frequency cepstral coefficient |
| MLP | multilayer perceptron |
| MSE | mean square error |
| OOV | out-of-vocabulary |
| PLP | Perceptual linear predictive analysis |
| POS | part-of-speech |
| RMS | root mean square |
| TWOL | two-level rules |
| TTS | text to speech |
| WER | word error rate |

# Abstract

Prosody describes the aspects of speech that go beyond its basic phonetic content. They include the melody and rhythm of speech, loudness and similar properties. Prosody is a vital part of speech that simplifies understanding and often serves as an additional information channel. While modern speech-synthesis systems are able to produce speech with prosody that sounds correct, they still lack the ability to make it lively enough so that the prosody does not immediately betray the artificial nature of the speech. This thesis aims to explore one the functions of prosody in synthesis, namely that of structuring speech. To that end, it proposes a more direct use of the syntax structure of the text for the generation of prosody.

A new hierarchic approach to prosody generation is introduced that produces prosody directly from the syntax structure of the sentence. This is achieved by defining elementary prosody generation functions, called mutators, that describe the local contribution of a single node in the syntax tree to the global prosody of the sentence. We show that these mutator functions can be trained from natural speech and then combined according to the syntax structure of the sentence for the prosody generation. The result is a complex, yet flexible prosody generation model that can produce natural sounding prosody. In contrast to existing methods, the mutator functions directly generate physical prosodic parameters like fundamental frequency and duration, although stylized on a word level. This means that abstract prosody concepts like accent and phrase are represented only indirectly in the model, making it easier to learn new prosody models from arbitrary examples of natural speech and allowing to express more fine-grained the degrees of prosodic expression. The thesis explores different word stylizations

and shows that they can capture the prosody sufficiently.

This thesis also describes in detail the process necessary to prepare natural speech data for the training of new prosody models using the example of audio book data. We discuss normalization of punctuation and text formatting for diverse text sources and efficient parsing of the texts. The thesis then presents an algorithm for phone segmentation of long speech recordings based entirely on forced alignment between text and speech using hidden Markov models. We show the algorithm to be mostly self-contained, requiring only a generalized language independent phone model that can be trained on any other available speech corpus, including on corpora in a different language.

This work concludes with a perceptual evaluation of six different prosody models trained on German and English audio books from the Librivox project. We show that the prosody produced by our model is preferred over a neutral prosody based on a simple accent/phrase model.

# Kurzfassung

Prosodie beschreibt den Teil der Sprache, der über den eigentlichen phonetischen Inhalt hinausgeht. Das beinhaltet Melodie und Rhythmus der Sprache, sowie Eigenschaften wie Lautheit. Prosodie macht einen wichtigen Teil der gesprochenen Sprache aus, der die Verständigung erleichtert und oft auch als sekundärer Informationskanal dient. Obwohl moderne Sprachsynthese-Systeme weitgehend in der Lage sind, korrekte Prosodie zu erzeugen, fehlt ihnen nach wie vor die Fähigkeit eine Lebendigkeit und Abwechslung zum Ausdruck zu bringen, die menschliche Sprache auszeichnet. Diese Arbeit beschäftigt sich mit einer der Funktionen von Prosodie für die Synthese, nämlich die gesprochene Sprache zu strukturieren. Dabei geht es darum, die Syntaxstruktur eines Textes direkter in die Generierung der Prosodie einfliessen zu lassen.

Es wird eine hierarchischer Ansatz zur Prosodie-Generierung entwickelt, mit dem Prosodie in direkter Relation zur Sytnaxstruktur des Satzes berechnet wird. Dazu werden elementare Funktionen definiert, sogenannte Mutationsfunktionen, die den lokalen Einfluss eines jeden Knotens im Syntaxbaum des Satzes auf die Prosodie beschreiben. Es wird gezeigt, dass die Mutationsfunktionen anhand natürlicher Sprache trainiert und dann kombiniert werden können, um die Prosodie komplexer Sätze zu erzeugen. Im Unterschied zu existierenden Methoden generieren die Mutationsfunktionen direkt physikalische Prosodieparameter des Sprachsignals wie Grundfrequenz und Lautdauer, jedoch auf Wordebene abstrahiert. Das bedeutet, dass abstrakte linguistische Konzepte wie Akzent und Phrase nur noch indirekt repräsentiert werden, wodurch das Lernen neuer prosodischer Ausdrucksweisen vereinfacht wird. Die Arbeit untersucht verschiedene Wordabstraktionen und Trainingsverfahren für die Mutationsfunktionen.

Desweiteren wird im Detail die automatische Aufbereitung von natürlichsprachlichem Trainingsmaterial im Allgemeinen und Audiobüchern im Speziellen behandelt. Es wird diskutiert, wie der Text bezüglich Satzzeichen und Formatierung normalisiert und dann effizient geparst werden kann. Die Arbeit präsentiert dann eine Methode zur phonetischen Segmentierung von langen Sprachaufnahmen, die vollständig auf Forced-Alignment mit Hidden-Markov-Modellen basiert. Die Methode benötigt nur einen kleinen externen Training-Corpus, der auch in einer anderen Sprache sein kann.

Die Arbeit schliesst mit einer subjektiven Evaluation von sechs verschiedenen Prosodie-Modellen, die an englischen und deutschen Audiobüchern des Librivox-Projekts trainiert worden. Es wird gezeigt, dass das entwickelte Model einem einfachen Phrasen/Akzent-basierten Model vorgezogen wird.

# Chapter 1

# Introduction

## 1.1 Problem Statement

Speech is a very complex form of human communication. In everyday use, we tend to think of speech foremost as a means to transfer a message by articulating a sequence of phones that form words which in turn form meaningful sentences. But speech is more than that. Beyond the words, there are secondary information channels that are equally important but that we are much less aware of. The tone of voice, the melody, the choice of emphasis and the clarity of articulation can all serve to give structure to speech, to enforce the meaning of the words and to convey additional information. From the way an utterance is spoken, the listener can determine the true intent of the speaker, like truthfulness or sarcasm, or infer his or her emotional state.

These properties of speech beyond the phonetic content of the utterances are generally combined under the term *prosody*. In the speech signal, it manifests itself in the melody, the rhythm and the intensity of speech. In other words, prosody describes *how* words are expressed, and as such, it is an essential part of speech. Without prosody, speech sounds artificial and robotic at best. As prosody gives an utterance structure and helps to disambiguate syntax where necessary, it helps to improve intelligibility [Sil93] and to avoid misunderstandings.

Nonetheless, prosody has escaped a clear and concise classification so far. While linguistics have been able to describe confined prosodic effects and analyze them in artificial environments, the interplay between prosodic parameters and the different influences is still little understood [Hir02]. Even between trained linguists there is a larger disagreement in the annotation of accents and phrases than of phonemes [SM00].

The lack of a well-defined description of prosody has been a particular challenge for the generation of natural sounding output in the field of speech synthesis. The generation of correct pronunciation has been largely mastered and there have been important improvements in the generation of close-to-human voice characteristics. Yet the prosodic characteristics still easily betray the synthetic nature of speech. It is not so much the complete lack of prosody or the presence of gross errors but the monotony and repetitiveness of the prosody in the generated sentences. Without applicable rules, state-of-the-art text-to-speech (TTS) systems rely mainly on data-driven methods to learn statistical models of prosody from natural data. These models in turn require properties to be extracted from the text that are relevant for prosody. The scope of these properties is still very limited, concentrating on properties of phones and syllables and making very little use of the wider context of the text. The result is a minimal form of prosody that sounds natural enough for a simple utterance in an isolated context but lacks the prosodic variations that could help the listener to put the sentence in a larger context of the discourse. As soon as longer sections of speech are synthesized, it becomes evident that all sentences follow a similar pattern, making listening an exhausting experience. To a certain degree, this still hampers the acceptance of TTS systems for some applications. They are already widely used in navigation or automated announcement systems where only few sentences are required that in addition have a very predictable structure. Similar observations can be made about the latest developments in personalized assistants like Apple's Siri which possesses a very limited speech repertoire. There are, however, other applications where longer rendering of speech with arbitrary structure and content is needed. Examples are extended dialog and information systems, or readers for personalized messaging systems or books in eyes-busy situations. They will only gain acceptance if prosody becomes more lively and carries the additional meaning found in human speech.

## Goals

Prosody is the product of very different influences: the speaker's mood, their attitude towards the listener, the content of the message, the dialog setting, to just name a few. The goal of this thesis is to develop a prosody generation algorithm that makes more extensive use of syntax information in order to capture nuances in the text that can be used to create a more varied prosody. The syntax structure of a sentence is one of the elements of a sentence that reflects at least partially the same influences that also govern the realization of prosody, in particular in terms of the structure of the dialog. While by no means comprehensive or complete, the syntax tree is much easier to obtain then a semantic analysis. Parsing techniques have been well researched, and efficient and fast algorithms are available. We aim to explore if the information obtained through the syntax analysis is sufficient to create prosody that allows to synthesize longer bodies of text without repetitive effects that make current state-of-the-art systems sound stereotypic.

The basis will be the SVOX TTS system [Tra95], a system that already incorporates syntax parsing as a means to include linguistic information for prosody generation. The SVOX system follows the conventional paradigm where first an abstract prosody description consisting of phrases and accents is created which is then further translated into a concrete realization of prosody. The goal of this thesis is to abolish this intermediate abstraction step and develop a prosody generation method that is able to directly infer the appropriate prosody realization from information obtained from the syntax tree. Such a system needs to follow a statistical approach and be able to learn the relation between syntax and prosody realization automatically.

The thesis will concentrate on one specific application of TTS systems: the reading of novels. Reading of literary texts has the interesting property that the TTS is in the same situation as a human reader in that only the text itself is available. Intention and meaning need to be inferred from this text alone. All information that is needed for a natural prosody rendering must therefore be available. Even though human readers have the advantage they can also interpret the meaning of the text, a synthesizer should be able to reconstruct much of the information through a syntactic analysis.

Audio books have the further advantage that human-read examples are readily available as training material for the statistical prosody models. They are generally of a sufficiently high quality that they can be used directly with little or no manual preprocessing. Nonetheless, they still need to be prepared for training. The development of methods for fast automatized processing was a secondary goal of this work and will be discussed in the second part of this thesis.

## 1.2 Scientific Contribution

The following contributions result from this thesis:

1. We have proposed a novel approach for prosody generation from syntactic information based on a hierarchical combination of statistical learners. In contrast to existing models, the approach does not require to select a priori the syntactic constructs that are relevant because the relevance is learned during training as well.

2. We have proposed to replace the conventionally used abstract prosody descriptions based on accents and phrases with a new stylization of prosody, an abstraction of prosodic signal parameters on word level. Such a stylization steps away from the paradigm of categorization of prosody, allowing a higher degree of freedom for the prosody generation process. We have shown that despite that the stylization can be generated from text in a way that is sufficient to generate the natural-sounding prosody.

3. We have developed a process to prepare large corpora from arbitrary sources for the training of this prosody generation model. The process only requires a grammar for the language, the audio recording and the text source. We have used this process to rapidly create models in different styles from different speakers as another step towards a more varied prosody expression.

4. We have proposed a new language-independent segmentation algorithm for large corpora. We have shown that phone alignment based entirely on forced alignment with hidden Markov models (HMMs) can be used on large corpora with the same reliability as speech recognizers which are conventionally used for this task.

## 1.3 Structure of this Thesis

This thesis is roughly divided in two parts. The first part describes hierarchical prosody generation and its application. The second part deals with preprocessing of large audio corpora that are required for the training of the prosody models.

**Chapter 2** explores the different definitions of prosody and defines the relationship between the syntax of written text and prosody. It concludes with an overview of existing prosody generation techniques in speech synthesis.

**Chapter 3** introduces the concept of direct prosody generation from a syntax tree and discusses the implications for the construction of a syntax grammar that is prosody-aware.

**Chapter 4** describes in detail the statistical algorithm employed for computing prosody directly from the syntax tree. It introduces the concepts of mutators and word prosody contours which are central to the prosody generation method.

**Chapter 5** gives an overview over the corpus preparation process used to prepare large amounts of training material from audio books and details improvements implemented in the SVOX parser to be able to handle long, continuous texts.

**Chapter 6** explains in detail phone segmentation for large texts, introducing a two-stage approach for segmentation that can be used almost autonomously.

**Chapter 7** discusses the application of the extended prosody generation to the task of reading books. It also presents the results of a perceptual evaluation of the prosody models.

**Chapter 8** provides some concluding remarks and an outlook for further work.

# Chapter 2

# Prosody and Speech Synthesis

Prosody in general covers a very wide range of research fields, including linguistics, acoustics and signal processing. In this chapter, we define the basic terms used in this thesis and put them in a context of related work. The first sections explore prosody from a linguistic point of view. The different abstraction levels will be defined and the relationship between written text and read speech studied. The final section gives an overview over existing TTS systems and how they approach the problem of generating prosody.

## 2.1 Definitions of Prosody

There is no single accepted definition for the term prosody. In a very broad sense, it may be defined as the sum of supra-segmental aspects of speech that is present as a secondary information channel, in contrast to the primary information that is transmitted through the words.

A more concrete definition depends on the level of abstraction with which speech is viewed. From a signal processing point of view, when speech is seen as an acoustic signal, prosody is defined by a set of measurable physical parameters of the signal. In particular, three features

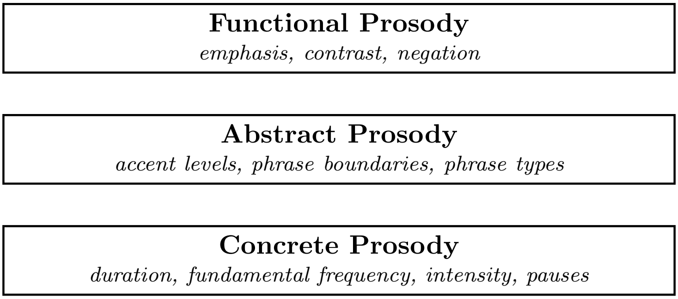

**Functional Prosody**
*emphasis, contrast, negation*
**Abstract Prosody**
*accent levels, phrase boundaries, phrase types*
**Concrete Prosody**
*duration, fundamental frequency, intensity, pauses*

Figure 2.1: *The abstraction levels of prosody.*

are considered to define the prosody of an utterance: the fundamental frequency ($F_0$), the duration of each phone and the loudness (or intensity) of the signal. On a purely linguistic level, when speech is considered as a means of communication, prosody is a function of human perception, interpreted in terms of its effect on the communication process.

In speech synthesis, both views are used and are often mixed up with each other. To avoid confusion of the different abstraction levels, the definitions as shown in figure 2.1 shall be used in this thesis.

*Functional prosody* shall describe the level of intent behind the prosody realization. This is the actual message the speaker tries to convey to the listeners, either consciously or unconsciously. On a meta-linguistic level, these functions may include for example emotions or sarcasm. Closer to syntax, and more relevant for this thesis, it includes functions to enforce meaning by expressing for example negation, contrast or emphasis.

*Concrete prosody* expresses the actual realization of prosodic parameters in the speech signal, in particular $F_0$ and duration. Note that these physical parameters are not only a product of functional prosody but also have a strong segmental component. They heavily depend on the nature of the concrete phoneme that is being spoken. For example, each phoneme has a natural length that determines for a large part the realized duration. Concrete prosody nonetheless shall refer to the final parameters applied to the speech signal. When segmental influences are removed, we shall use the term *stylized prosody.*

Finally, *abstract prosody* represents an intermediate level that also describes a kind of prosody realization but in terms of human perception. The basic elements are phrases (the grouping of speech) and accents (the degree of emphasis for each syllable). Sometimes, different phrase types (e.g. statement type, question type) are distinguished as well. Abstract prosody is frequently used as an intermediate level of description and has influences from concrete prosody (e.g. pitch accents explicitly refer to a change in $F_0$ in an emphasized syllable) and functional prosody (e.g. emphatic accents which refer to the function of special emphasis).

## 2.2 Functional Prosody in Speech

Prosody in a functional sense gives the text a structure and meaning beyond the basic lexical semantic of the simple words. However, that is not the only purpose of prosody. It is the product of many different functions that overlay each other.

First of all, prosody has a structuring function. Phrasing organizes speech into smaller chunks of information that help the listener to process the speech. Phrasing is also used to disambiguate syntax structures. Prosody allows to mark prominence in sentences which helps to distinguish between new and given information, between topic and comment, thus introducing discourse structure. In addition, emphatic emphasis may be used to single out certain words or even entire phrases. The intonation may also mark special constructs like reported speech.

Intonation also expresses the intent of the speaker, explicitly or implicitly. It may explicitly support the grammatical structure of the sentence, as in interrogative or imperative sentences but also in structures like enumerations or negations. Implicitly, ambivalence between prosody and meaning or syntax can be used to transmit a hidden meaning. A prominent example of hidden intent are irony or sarcasm, where prosody may be used to mark that the speaker means the opposite of what the words are expressing. On syntax level, an example are grammatical questions that are rendered using the prosody of a statement sentence. Depending on language habits, this may indicate a polite order or suggestion.

Finally, emotions have a strong influence on the prosody, so that the emotional state of the speaker may be transmitted through prosody to the listener.

Functions related to phrasing and prominence are structural in nature and can to a certain degree be determined from the syntax structure of the text. They are the part of functional prosody the following sections will focus on. The other functions mentioned above depend entirely on the interpretation of the speaker. They will be discussed shortly in chapter 7.

## 2.3 From Written Text to Spoken Prosody

When spontaneous speech is produced by a human, the speaker is generally aware of the intentions behind the utterance and can unconsciously adopt the prosody accordingly. The main purpose of a TTS, however, is not the production of spontaneous speech but the reading of potentially arbitrary texts. This is a very different process even for a human speaker. It entails the interpretation of sentences produced at a different time and potentially by a different person. The reader thus is in the same situation as a TTS system in the sense that only the written text is available and the intentions behind it need to be inferred from what was written. The result is a prosody that is audibly different from that of spontaneous speech. For the construction of a TTS it is important to understand what information a human reader uses when reading a text aloud, so that the same information can be extracted and used in the prosody generation process. The following two sections will therefore explore the relation between written text and its realization in terms of prosody.

Studies on reading comprehension ([Ken12] and references therein) have found evidence that the syntax structure plays an important role in the decoding process but there is no agreement to what degree it is used in comparison to the meaning of the sentence. Structural precedence theory asserts that syntax structure is determined first and independently of meaning. When the reader of a text uses prosody to simplify comprehension for the listener, the syntax structure must have a strong influence on the spoken rendering of the text. [KKG02] even asserted that the applied prosody is primarily a result of the structure of the text while meaning only plays a minor role. Experiments with placing of phrase boundaries [GE72] indicate that this is even more true for read speech than for spontaneous speech. However, these studies have focused on isolated sentences and are therefore only applicable for the production of neutral prosody outside the context of a longer discourse. To make longer texts truly comprehensible, the context of the sentence cannot completely be dismissed.

A comprehensive study on the interaction between speaker intentions and listener comprehension within a discourse can be found in [Cha94]. Chafes establishes that phrasing and accentuation are a means to order information that is transferred from the speaker to the listener. Intonation phrases represent a single piece of information and are sized such that they can be retained in memory. The type of accentuation varies according to how well known the piece of information is assumed to be known by the listener. A common distinction is between new and given information. Chafe further distinguishes how easily it can be activated.

Written language needs to follow similar principles of discourse structure to be easily comprehensible to the reader. Chafe [Cha88] even went as far as asserting that text is produced by the writer with a very specific reading in mind and structured accordingly. If a synthesis system is able to detect these structures and learn how to translate them into equivalent spoken prosody, more natural sounding prosody can be produced. [**?**] even argued that the intonation structure of a text can be parsed just like the syntactic structure. He proposes to use a combinatorial contextual grammar where context information like newness is encoded on a lexical level and the constraints of the grammar ensure correct bracketing according the information structure of the text.

In the following section, those structures that can be determined from the text without using a complex semantic analysis will be examined in more detail.

## 2.4 Prosodic Hints in Writing

There is only one direct equivalent to prosodic expression in written language, which is punctuation. If a writer wants to convey meta-semantic information beyond that to the reader, stylistic means need to be used. In this section, we first discuss in detail the role of punctuation and then shortly introduce two other means to hint the intended functional prosody: choice of syntax and choice of words.

### 2.4.1 Punctuation

Punctuation is the most important syntactic marker for a prosody generation system because it functions as an explicit prosody marker in written text. Punctuation marks almost always coincide with an intonation phrase boundary, thus imposing a basic phrase structure on the read sentence. In addition, they may be used to express phrase types and, to a lesser degree, hint the intent or emotional state of the speaker. Especially when used in the latter sense, punctuation marks may include symbols beyond simple punctuation marks in a grammatical sense. Therefore, punctuation shall in the following refer to any markers in the text that are not lexical words.

While the use of punctuation is governed to a certain degree by the official grammar and style rules of each language, writers still have a relatively high degree of freedom in their application and might even consciously ignore established conventions to achieve a certain stylistic effect. This is even more true for texts from the previous centuries when language rules were applied much more liberally. As an example, take the following excerpt from *Emma* [Aus]:

> “Oh yes!—that is, no—I do not know—but I believe he has read a good deal—but not what you would think any thing of. He reads the Agricultural Reports, and some other books that lay in one of the window seats—but he reads all *them* to himself. But sometimes of an evening, before we went to cards, he would read something aloud out of the Elegant Extracts, very entertaining. [. . . ]”

Here the author uses punctuation in a stylistic sense to mimic spoken language more closely. It can be seen that punctuation is not only used to emphasize the intended phrasing but it also gives an indication of the emotional state of the speaker. The first exclamation *Oh yes* clearly marks excitement, the dashes express agitation or hesitation. A similar use of punctuation can often be found to mark sarcasm, rhetoric questions, etc.

The example also contains another form of prosodic marking only loosely related to punctuation: emphatic markers in the form of a different font style. The normally unaccented *them* is elevated to strongly accentuated state by the use of the italic font style. Although font markers are much more rare, they can also be a useful indicator of the authors intentions.

### 2.4.2 Importance of Syntax

The grammar of a language already functions as a constraint that forces specific syntax structures for many of the discourse-related aspects of functional prosody. For example, sub-ordinate clauses can be distinguished from main clauses through syntax only and many languages even further differentiate multiple types of sub-ordinate clauses. Similarly, negative expressions, enumerations and comparisons can often be identified from the syntax structure only. A more complete analysis of structures that are important in terms of prosody will follow in 3.2.2.

Next to the mandatory use of appropriate grammatical constructs, the grammar rules allow a certain freedom in their application. This allows to use different syntax structures to express the same meaning but with a slightly different connotation, similar to what can be achieved with prosody.

The most important example of such a shift in syntax structure is a change in word order in the sentence. This may be used to change the emphasis, in particular when introducing new concepts, or to establish a relation with previous sentences. Even languages with a strict word order like English allow to employ such constructs. An example is the sentence

> “It was Miss Taylor’s loss which first brought grief.”

again taken from *Emma*.

In the context of prosody generation it is important that such constructs can be parsed, even though they may appear much less frequently than sentences in standard word order.

### 2.4.3 Choice of Words

As with syntax, the lexical repertoire of a language leaves the writer the possibility to express connotations that a speaker might prefer to express via prosody. For content words these are mostly synonyms with a slightly different meaning. Normally the word choice sufficiently expresses the desired function, so that it is rarely reflected in the realization of concrete prosody[1]. The use of words in general, however, also determines the flow of information in the discourse. While important for the prosody rendering, a proper detection requires a semantic analysis of the text which is outside the scope of this thesis.

## 2.5 Prosody in TTS Systems

The remainder of this chapter will be about the speech processing aspect of prosody generation and discuss existing approaches. The purpose of the prosody generation part in a TTS system is to compute the concrete prosody realization for a written text. This is always realized as a two stage process. First, the text is analyzed and certain features are derived that are considered relevant for prosody generation. The exact nature of these features varies between different synthesis systems, as we will see in the next section. We shall nonetheless combine them under the generic term *prosodic properties*. This extraction normally happens together with the generation of the pronunciation for the text, so that the result of this first step of text analysis is a sequence of phonemes annotated with the prosodic properties. In the second step, this description is used to generate the concrete prosody which is later used for the synthesis of the speech signal.

[1] An exception to this rule are words that are emotionally charged. Compare the phrases “He bought a large house.” and “He bought a giant mansion.”. A human reader might be tempted to add a special emphasis to the adjective in the second sentence.

The different means of prosody generation that have been proposed over the years can be classified by whether they used a rule-based or a statistical approach, and by the type of prosodic properties used. In the following, we will first discuss the different prosodic properties that have been proposed as the output of the text analysis and then continue with an overview of prosody generation methods. The chapter will finish with a detailed introduction of the SVOX TTS which is the basis for the work in this thesis.

### 2.5.1 Prosodic Properties

Prosodic properties have been proposed on very different levels. Most commonly the properties are based on an abstract prosody description with accents and phrases as their base elements. Phrases are the basic intonation units which are delimited by audible boundaries. They are characterized by their type (e.g. question, statement) and by the strength of the surrounding boundaries. Accents mark the focus of each phrase or of the entire sentence. They are also characterized by their strength (normally relative to each other) and sometimes by their realization (e.g. pitch accent) or function (emphatic accent). This model has the advantage that it is independent from speaker realizations. However, it is not able to account for more fine-grained changes in prosody and the annotation. The other disadvantage of this model is that the assignment of accent and boundary strength is, at least partially, subjective, which makes it difficult to obtain a consistent annotation for natural speech. As such an annotation is often necessary to create training material for data-driven approaches, other models have been developed that are closer to the realization of concrete prosody.

In her thesis, Pierrehumbert has developed an intonation system that allows to categorize typical $F_0$ movements [Pie80]. This work was later extended with phrase boundary types, leading to the creation of the TOBi system [SBP$^+$92] which is still widely used as an abstract prosody description in speech synthesis systems. A more recent adaptation of Pierrehumbert's work is the Rhythm and Pitch (RaP) system by Dilley [DB05]. Its goal is to annotate perceptual impressions instead of physical events, for which they report a higher inter-labeler agreement [DBG$^+$06]. However, the implications on the quality of TTS systems have not yet been tested.

Even closer to concrete prosody, the TILT model [Tay00] is a direct stylization of $F_0$ and duration on syllable level, that has also been used as abstract prosodic properties, although more frequently the representation is generated from abstract prosody [TB94]. Its parameters include measured start $F_0$, amplitude, tilt and peak $F_0$ position and duration.

Finally, a rather different approach was presented in [KS03] with Stem-ML, a mark-up language for $F_0$. The tags of this language define a mixture of a superpositional and template model. Phrases are described as parametrized curves on which accents are overlaid in form of template curves. Kochanski claims that these templates are motivated by physiological terms making them in principle language independent and creating a smooth $F_0$ contour. He assumes that prosody planing is confined to prosodic phrases, that there is no influence on $F_0$ between phrases, which is why the two levels of description are sufficient.

### 2.5.2 Extraction of Prosodic Properties from Text

Rule-based approaches have been dominant in earlier proposals for obtaining the prosodic annotation. Most of these systems expect the text to be parsed first and then use the resulting syntax tree for the prosodic analysis. The methods can be grouped into structure-based and constituent-based methods.

The first group of algorithms only uses the structure of the syntax tree to algorithmically derive abstract prosody parameters and has been mainly been proposed for deriving phrase boundaries (e.g. [Bie66], [Sel80], [Alt87], [FB89]) or accent annotation (e.g. [Kip66]). Structural parameters are for example the depth of the tree or the closeness of tree nodes in terms of the closest common ancestor. They are in principle language-independent but nonetheless have been developed with specific languages in mind. It remains unclear in how far different languages follow the same principles for phrasing and accentuation, a necessary prerequisite to justify a transfer between languages.

Constituent-based methods also take into account constituent information. They look at the type of nodes in the tree and develop transformation rules that are applied only to specific contexts. An example is Bachenko's algorithm [BF90] which uses rules that rely on constituents, constituent length and adjacency to a verb. To derive accent strengths, Kiparski's nuclear stress rule [Kip66] recursively reduces accents in subtrees according to whether they constitute the nucleus of the constituent or not.

These systems are generally well suited to cover what can be best described as *neutral prosody*. That is a prosody that sounds reasonably balanced in isolation but becomes repetitive for longer sections of synthesized speech because the prosodic patterns that are produced are very stereotypic. Structure-based approaches in particular do not take into account special language constructs. The main disadvantage, however, is that the construction of rules is relatively labor intensive. Both approaches always need to be tuned towards one specific language and while they may still be usable for closely related languages, they are by no means language-independent. Even more labor intensive are constituent-based approaches where the rules have to be adapted to the specific grammar that has been used to create the syntax trees.

For these reasons prosody generation in state-of-the-art systems uses mainly data-driven methods. The use of properties derived from the syntax structure has diminished in these approaches. One of the few exceptions is a method by Hirschberg et al. where pitch accents and phrasing are learned from a tree representation of the sentence [HR01]. They showed that it can reach a better accuracy than flat feature sets as described in the next paragraph. Wang later analyzed the suitability of various syntactic features for the estimation of phrase boundaries using part-of-speech, word position, sentence length and simple constituent features and found part-of speech features and word-based distance measures to be the most important ones [WH92].

More prevalent in current TTS systems is the use of features that can be directly derived from the text without any intermediate syntactic analysis. The usage of linguistic information is limited to annotations like part-of-speech that may be contained in the pronunciation dictionary. Features describe word, syllable and phone by their class and position [Str02]. The actual selection of feature classes is done manually.

While the annotation can be largely automatized with the data-driven methods, the features used cover only a relatively small context within the text. They are not able to describe structural dependencies between neighboring sentences or sentence parts. The result is a sometimes slightly arbitrary prosody with obvious local errors.

### 2.5.3 Generation of Concrete Prosody

Concrete prosody is so diverse in its appearance that the development of rule-based systems is an extremely complex task, which is why statistical models are predominant for the generation of the final prosody parameters. One of the few proposals for rule-based computation is an algorithm by Anderson [APL84] which translates Pierrehumbert's $F_0$ intonation system into $F_0$ contours.

Statistical models applied to $F_0$ and duration generation include linear regression models ([Bla96], [vS94]), CART trees ([MDM98], [RO99]), multivariate adaptive regression splines ([Rie97]), decision trees ([Ril90], [YKK08]), neural networks ([Rie95], [Rom09b]) and support vector machines ([LMG$^+$11]). No significant difference in performance has been reported for any of these models. When comparing duration modeling for HMM synthesis, Silen found a slight advantage for regression models over CART models [SHNG10] but these tests were restricted to objective RMSE comparison only.

With the introduction of concatenative synthesis, in particular unit-selection [HB96], corpus-based modeling of prosody was introduced. The idea here is to use natural prosody from a large speech database with as little modification as possible. Such unit-selection systems sound very natural if long enough matching chunks are available. They are therefore still of great interest in closed-domain systems with a limited, predictable output. The main disadvantage is that the output is bound to the prosody of the underlying database. Therefore, a more varied and lively prosody can only be achieved by enlarging the available speech samples which can become very costly in terms of memory and production effort. A variant of this approach is the use of intonation patterns from natural speech [MDM98] where $F_0$ is synthesized with a standard unit-selection method and then implanted on synthetic speech.

HMM-based systems finally unify the generation of spectral and prosodic speech features. Duration and pitch are modeled together with the spectral features of speech [YTM$^+$99]. Pitch is represented as a feature of the generative HMM while duration is represented through the state durations which are modeled as Gaussian distributions. The prosody is determined by the choice of the context-dependent HMM which is chosen via a decision tree. The contextual factors include phoneme characteristics, part of speech, phrasal context and accent.

### 2.5.4 Hierarchic Models

The models discussed so far have in common that prosody is produced as a single linear sequence from a prosodic description that is a linear sequence as well. It has, however, been long discussed in linguistics that prosody is composed of multiple tiers, e.g. phonetic, foot, phrase and sentence level (e.g. [Sel86]). On these grounds hierarchic prosody models have been proposed that view prosody as a combination of basic contours from these different levels.

One of the first hierarchic models was the Fujisaki model [Fuj88], a superpositional model composed of exactly two levels: a phrase and an accent level. Both are modeled as impulse response models that are additively combined. Synthesis systems using this model have been implemented in many languages, e.g. [MJ01] for German, [FO98] for the multi-lingual case.

The ProSynth model [OHH$^+$00] is a rule-based system that parses the text for phonological structures (intonation phrase, accent group, foot, syllable) and then applies a number of transformation rules to obtain the concrete prosody. Stem-ML [KS03] mentioned previously also represents a hierarchic model where phrase and accent level are generated by templates. Finally, the Phrase/Accent model by [AOB11] uses a TILT-based modeling for contours at accent level onto which a mean $F_0$ of the phrase is added.

In most models, the base contours are related to prosodic units: syllable, foot, phrase. The prosodic units form a hierarchy much like the syntax tree. The main difference is that the structure is motivated by prosodic events. An approach where the units are much closer to the syntax structure is the Superposition of Functional Contours (SFC) model proposed in [BH05]. The model distinguishes between syllable and segment prosody generation. The hierarchical description that is used for the syllable model is created by transferring the syntax tree into a sequence of *chunks* with hierarchical dependency relations that express the syntax and discourse structure of the sentence. This transformation and reduction process from syntax tree to chunks is rule-based. The syllable prosody is then used in the statistical models for the generation of the segmental prosody. The model of multiple unit sequences [vSKKM05] is another additive model which is similar to the SFC model. The main difference is that it uses a combination of natural prosody curves instead of computing the concrete prosody with statistical models.

### 2.5.5 The SVOX Synthesis System

SVOX [Tra95] is a text-to-speech system that was developed at ETH with the premise to include as much linguistic information as possible in the synthesis process. It was later extended to the PolySVOX system [Rom09a] which supports synthesis of multiple languages in parallel and can also produce mixed-lingual sentences. This latter system has been used as the basis for the work in this thesis.

SVOX can be roughly divided into a voice-independent text processing part and a voice-dependent signal synthesis part. The prosodic properties that function as an interface between the two are encoded in an abstract *phonological transcription*, the phoneme sequence enriched with word and syllable boundaries, accents, phrase boundaries and phrase types. For example,

```
#{P:0} \G\di \E\[2]e_@r f[2]O:s w[1]Vn
#{T:4} \G\l[3]an-d@-t@ ?In fr[1]aNk-fUrt
```

represents a phonological transcription of the German “Die Air Force 1 landete in Frankfurt.” (Air Force 1 landed in Frankfurt.) where `#{X:N}` is a phrase boundary of strength `N` introducing a phrase of type `X`[2], `\G\` and `\E\` denote language switches to German and English respectively and `[A]` is an accent of strength `A`.

The text processing part is completely rule-based and can be further divided into parsing of the text and a phonological processing stage which transforms the syntax tree into the phonological transcription.

[2] The original SVOX implementation can handle exactly two types: progressive (P) and terminal (T) phrase. PolySVOX extended the phrase type set by four more types: progressive and terminal question phrase, simple statement, exclamation.

The signal processing part uses data-driven methods and is split into the prosody control stage, where the concrete prosody is computed using the prosodic properties from the phonological transcription, and the speech signal generation. Figure 2.2 depicts the complete SVOX system.

The **text analysis** is realized with a definite clause grammar (DCG) parser. There are mono-lingual dictionaries and grammars for each language. When loaded at the same time, the parser can analyze the text in multiple languages in parallel and determine automatically the correct language of the input text by determining which is the highest ranked parse result. Language switching within a sentence is also possible. To that end, small *inclusion grammars* are added that describe where in the sentence such a switch is possible. PolySVOX fully supports German and English and also contains a rudimentary dictionary and grammar for French suitable for small inclusions. The resulting syntax tree describes the sentence structure where words constitute the leaves. They are annotated with the orthographic and phonetic transcription of the word. Words themselves are parsed as well but the results are not forwarded to later stages.

The **phonological processing** stage transforms the syntax tree into a phonological transcription. This transformation consists first of all of *phonetic processing* which includes syllabification, morphological and phono-tactical transformations of the lexical phonetic transcription and transformation of word stress (mainly required for compounds). The syllabification is realized by parsing the words using syllable grammar. The other two parts are implemented via multi-context two-level rules [RP04]. They are standard two-level rules that are applied to subtrees of the syntax tree according to the context they appear in. An example is shown in figure 2.3. The other part of phonological processing, *accentuation and phrasing*, adds prosodic information. The determination of phrase boundaries follows the algorithm by Bierwisch [Bie66], which was originally developed for German only but is also applicable for other languages provided the syntax trees have a similar structure. Phrase types are then assigned according to the phrase types that make up the sentence. Accentuation implements the algorithms by Selkirk [Sel80] for French and Kiparsky [Kip66] for German and English. These algorithms are purely structure-based methods. To account for

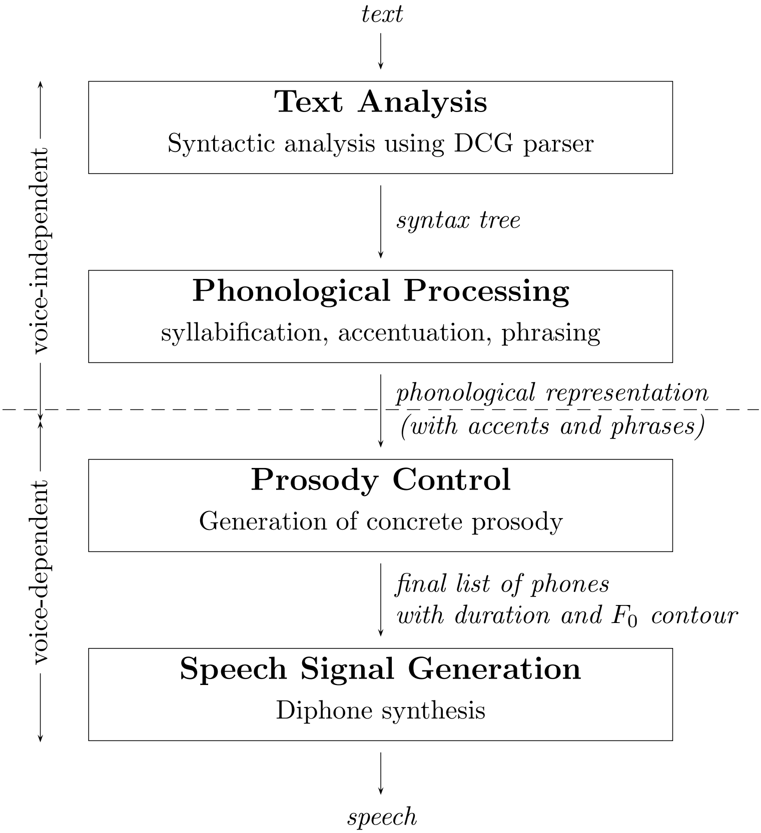

text
Text Analysis
Syntactic analysis using DCG parser
syntax tree
Phonological Processing
syllabification, accentuation, phrasing
phonological representation
(with accents and phrases)
Prosody Control
Generation of concrete prosody
final list of phones
with duration and $F_0$ contour
Speech Signal Generation
Diphone synthesis
speech
voice-independent
voice-dependent

Figure 2.2: *Overview over the architecture of the classic SVOX TTS system.*

```
N_G

'm'/@ <=> _ '-' < %LexAcc > 'm'
```

```
\G\'?a:t@m-'mask@-    ⟹    \G\'?a:t@-'mask@-
```

Figure 2.3: *Example application of multi-context rules. In the box, the* assimilation rule *described by the two-level rule below is applied only to subtrees that match the context above (i.e. noun words). For the German noun* Atemmaske*, the rule would match the subtree containing the word only and remove the 'm' before the syllable boundary.*

non-standard stress patterns and other language particularities, additional accentuation patterns are applied. They specify modifications of the accents for particular subtree configurations. These rules are language-specific and need to be adapted manually to the specific rules of each grammar.

The **prosody control** stage generates the concrete $F_0$ for each syllable and the segment durations for each phone from the phonological transcription. This stage is realized with two independent artificial neural networks (ANNs). The features for the networks are extracted from the phonological transcription. Original SVOX took the view that the context described by the features can remain relatively narrow because the phonological transcription itself was created taking into account the entire structure of the syntax tree. Consequently, the ANNs for the original SVOX remain relatively small and very little training material was required for the development of a new voice. PolySVOX deviates from this view, mainly to avoid making any assumptions about the type of features that influence the prosody and thus be able to maintain a feature set that is language-independent. Consequently, it includes features from a much larger context and also extends the types of features extracted from the phonological transcription. To improve the learning process and obtain a feature selection, weighted ensemble ANNs were used [Rom09b].

Finally, **speech signal generation** outputs the speech signal and is realized as a diphone-based synthesizer. It expects a list of phones as input, their length and the $F_0$ contour as interpolation points.

Each stage is realized independently with a well defined data interface between them, so that they can be easily replaced.

## 2.6 Discussion

In this chapter, we have started with a general definition of prosody and shortly discussed the factors that influence the realization of concrete prosody. We have also shown that written text contains many hints on the prosody that are reflected in the syntax structure and thus can be extracted using well-established parsing methods.

A survey of TTS systems shows that state-of-the-art systems use such hints very sparsely. Rule-based systems are most promising in that they provide the mechanisms to take into account syntax-related information from syntax trees. However, the creation of hand-crafted rules for prosody proved labor-intensive and given our limited knowledge on the relationship between syntax and prosody involves too much trial-and-error to allow a systematic approach. The SVOX phonological processing stage is no exception to that. While the general accentuation and phrasing algorithms produce a consistent phonological transcription that allows to render prosody that is accepted as close to human, it is still somewhat repetitive with little distinction between different syntactic constructs unless the context-specific transformation rules are applied. However, only few of these rules have ever been realized in practice, precisely because of the difficulties in writing and testing them. What is more, the rules are tightly coupled to the structure of the syntax tree. Therefore, any change in the syntax grammar rules requires to revisit the prosodic rules as well.

Another limiting factor is the expressiveness of the prosodic properties. When an abstract prosodic transcription is used, the concrete prosody generation is constrained by the expressiveness of the level of abstraction chosen. When only a few accent and phrase types are available, only a limited number of prosody patterns can be generated. While it is always possible to extend the description and include more prosodic properties, such a strategy would soon lead to an explosion of prosody generation rules and make the creation of rule-based systems even more costly.

What is needed is a more direct way to construct a system that translates the syntax structures of a text into the appropriate concrete prosody without abstracting the prosody on an intermediate level. Given the limited knowledge we currently have about the relationship between the two and the complexity such a system has to incorporate, such a system should be able to learn the prosody representation from natural speech. In the next chapter, a concept of such a model will be introduced that uses the information about functional prosody inherently present in the syntax structure of the text and translates that to concrete prosody.

# Chapter 3

# From Syntax to Prosody

The grammar of a natural language is not a logical set of rules that is applied deterministically to speech by humans. The grammar can rather be considered a loose collection of commonly agreed-on guidelines. For every rule, there are exceptions and legal violations that when used can subtly change the meaning. Consequently, a rich variety of syntactic structures can be found in everyday speech making the syntax analysis of a text no less complex than the prosody production process. While the analysis process could be simplified by concentrating on a smaller subset of the possible syntactic structures, it is exactly that diversity of the syntax structure that needs to be exploited if we want to obtain a prosody with more subtle changes and variations than what current TTS systems can achieve.

In this chapter, the general concept of a syntax-driven prosody generation method is introduced that aims to include the entire syntax tree in the process of generating prosody. Its goal is to use data-driven methods to learn to infer prosody directly from the syntax structures of a sentence. The second part of the chapter focuses on the structure of the syntax trees themselves, exploring in detail the requirements for the grammar to be able to generate syntax tress that properly reflect the functional prosody and therefore simplify the learning process.

## 3.1 Syntax-driven Prosody Generation

State-of-the-art TTS generally reduce the complexity of the task of prosody generation by restricting themselves to a very limited set of factors in terms of prosody. As we have seen in section 2.5.2, features mainly describe phonetic and syllable properties as well as linguistic properties on word level. They need to make some initial assumptions about what factors are the most important. The primary goal of the system presented here is to not only take the entire synax tree into consideration as a whole but also to make less prior assumptions about what configurations of the tree have an influence on the prosody. That means that the model has to learn during the training process, what features of the syntax tree are relevant for prosody. The advantage of such an approach is that explicit knowledge about the concrete prosody realization is no longer required, meaning that the model does not need to be tuned for specific speakers or languages.

This freedom comes at the price of an increased complexity of the prosody model. This section introduces the two basic concepts necessary to handle the complexity imposed by the syntax tree as input on one side and the concrete prosody as output on the other and shows how these concepts can be realized in the SVOX TTS.

### 3.1.1 Hierarchic Decomposition of Syntax

Training a statistical model from a syntax tree is not a straightforward task. First of all, there is no direct correspondence between the syntax structures and the concrete prosody. Not every constituent or constituent combination in a syntax tree is relevant for the prosody realization of the sentence. Therefore, the training process must include a feature selection process that is able to learn which configurations of the tree are relevant and which can be ignored. Conversely, there are aspects of the concrete prosody that are not reflected by the syntax structure and therefore cannot be learned by a purely syntax-driven model. Consequently, the concrete prosody represents noisy training data which can be difficult for the training of statistical models.

The second aspect that needs to be taken into account with respect to the syntax of the text is the complexity of the syntax tree. Statistical models always expect to operate on feature sets of a fixed size. Syntax trees, however, can vary greatly in complexity and size and cannot simply be transformed into a fixed set of features. What is more, the frequency with which a specific syntactic construct is used can vary widely. The goal of more liveliness can only be achieved if rare syntactic constructs are learned just as well as frequently used ones.

To be able to cope with both the complexity and the sparsity, the syntax-driven prosody generation borrows the idea of composed prosody contours from hierarchic prosody models. As with these approaches, we assume that the final concrete prosody is the result of the composition of overlapping *base prosody contours* (or base contours for short). We assume that each node in the syntax tree contributes such a base contour for exactly the part of the sentence that is covered by its subtree. That means that the overall prosody of the sentence is composed of base contours with an increasing scope reflecting the syntax structure of the sentence: there are base contours for each word, word group, phrase and clause, up to a base contour that covers the entire sentence. If it is further assumed that the local context of each node within the syntax tree is sufficient to determine its base contour, then the feature space for each node is confined enough that each base contour can be described with conventional statistical models with a fixed number of input features. The overall prosody model then is the collection of a number of such statistical models for base contours that are combined for each sentence according to its syntax structure. This way, complex sentence prosody models can be built from a relatively small number of simple models.

It is worth noting that the idea of prosody contours follows very closely the concept of the prosodic skeletons of the SFC model, briefly introduced in section 2.5.4. However, while in the SFC model the syntax trees are explicitly transferred into a structure that is closer to the intonation structure, we propose to use the syntactic structure directly, thus implicitly learning the relation between syntax and intonation structure.

### 3.1.2 Learning Concrete Prosody

In the last chapter, we have discussed the intermediate abstract prosody descriptions conventionally used for prosody generation, and we have seen existing approaches to learn abstract prosody like accents and phrases directly from the syntax. The hierarchic syntax-driven approach could learn to produce such abstractions, however, this has two important disadvantages: first of all, any data-driven approach to learning linguistically motivated abstract prosody requires appropriately annotated training data. As abstract prosody needs to be annotated by hand, this can easily be as labor-intensive as the development of a rule-based generation system. Second, and more importantly, relying on abstract prosody means that the prosodic properties taken into account are defined and fixed in advance. Therefore such a system would be limited in what can be expressed in terms of prosodic patterns and in the flexibility to adapt to different speaking styles or languages that may require different prosodic properties. A similar observation was made in [**?**] where it was called the quantisation error of the intermediate phonological level.

Both problems can be circumvented when concrete prosody is directly inferred from the syntax structures without the intermediate step of computing abstract prosody. Concrete prosody parameters like $F_0$ and duration can be computed directly and objectively from the speech material, allowing to automatically annotate any training material necessary. Any abstract prosody concepts are directly expressed in form of their effect on the concrete prosody. For example, it is no longer necessary to know how important is is to distinguish whether an accent is realized with elevated pitch or through lengthening. Indeed, it is not even necessary to distinguish between accented and unaccented words. Instead it is sufficient to annotate the lexical stress in the dictionary (if applicable) and the prosody model will realize the accent according to the model learned from the data.[1] This also has a direct effect on how prosody needs to be viewed when developing prosody models. To include a new prosodic realization that is known to be present in the training material for the prosody model but not realized in the synthesized prosody, it has to be described only in terms of its syntactic construct, generally by including explicit rules or attributes for it in

[1] Note that even though we abandon the linguistic prosody concepts like accents and phrases for the prosody generation at this point, these concepts are still realized indirectly through changes in the concrete prosody. The theoretical concepts that govern their appearance apply to the syntax-driven prosody generation just as to any other model. Therefore, we shall continue to refer to these concepts from time to time, even though their concrete realization in the model is meant, in particular, when discussing the relevance of syntax structures.

the grammar, as will be discussed in the next section. Most of the time this is a much easier task than stating the effect in terms of abstract or concrete prosody.

### 3.1.3 Statistical Prosody Generation for SVOX

The SVOX TTS system is an ideal basis for the implementation of a syntax-driven prosody generation as introduced in the last section. It already possesses a simple but powerful enough component for text analysis and the signal synthesis module is able to produce speech using concrete prosody values directly. Between these two components the statistical prosody generation will be implanted as shown in figure 3.1.

The parsing step remains largely unmodified. The text is first processed by the DCG parser which creates a syntax tree for each sentence with a lexical pronunciation at each word leaf. The tree is then passed on to a reduced version of the phonological processing stage that only executes syllabification, stress transformation and phono-tactical transformations of the phonetic transcription. Some of the existing phono-tactical rules require that the phrase boundaries are already annotated. For example, some assimilation phenomena in German do not occur on phrase boundaries. These rules could be rewritten to use grammatical phrases instead (noun, adjective or adverbial phrases). The multi-context rules provided an appropriate mechanism for that. Otherwise the rules for these processing steps could be used without modifications.

The output of the phonological processing is thus a syntax tree where the leafs contain words with their final pronunciation, lexical stress (if applicable for the language) and syllable boundaries. This tree is the input for the syntax-driven prosody generation. The structure of the tree is analyzed and base prosody contours are created and combined according to the syntactical structure. This is the *tree-derived model* of the prosody generation. At the beginning of chapter 2, it was already discussed that the concrete prosody has a strong segmental component, that is, it depends on the phonetic content of the words. The tree structure, however, does not carry any information about the phonetics of the sentence. The content of the words is independent of the syntax of the sentence[2]. Therefore, the prosody produced at this

[2] To be precise, there are restrictions on the words in terms of part-of-speech but

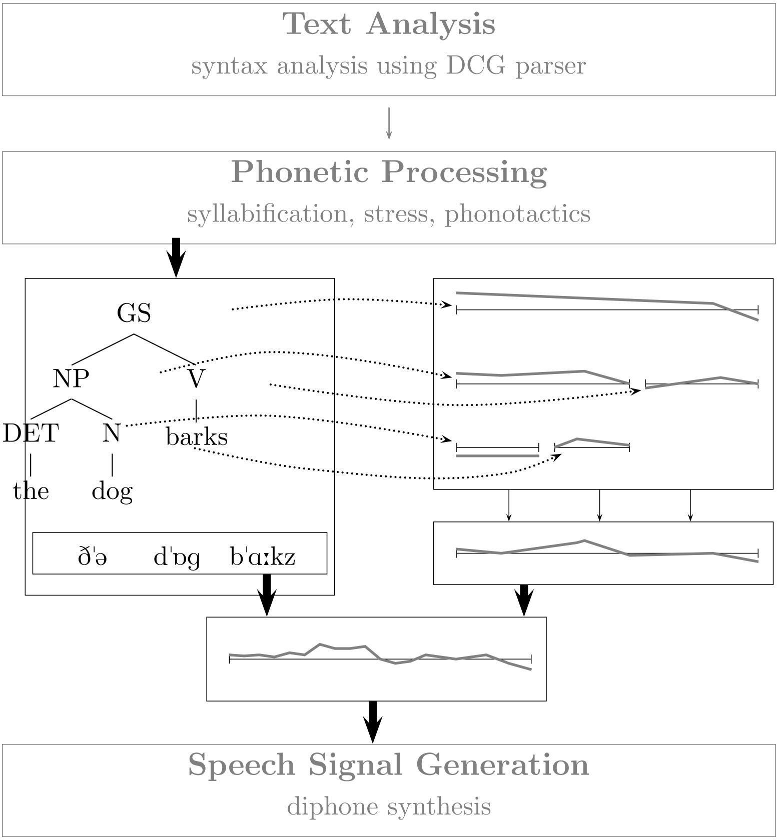


Figure 3.1: *SVOX architecture with statistical prosody generation. Each node in the syntax tree produces a basic prosody contour which is combined into a stylistic contour. This in turn is combined with segmental features to produce the final concrete prosody.*

point can only be a stylized prosody which describes the general trend of the prosody within the word independently of its content. This special stylization shall be called *word prosody contour* in the following. To obtain the final concrete prosody, the segmental information from the word leafs of the tree needs to be combined with the word prosody contour to compute the concrete prosody parameter for each phone. This final part constitutes the *segmental model* of the prosody generation.

When compared with the original SVOX architecture (c.f. figure 2.2) it can be said that the statistical tree-derived model replaces the rule-based accentuation and phrasing part of the system. The segmental model has the same function as the prosody control stage in the original SVOX. The phonological transcription remains the interface between the two stages with the small but important difference that the elements that describe abstract prosody concepts (accents, phrases) are replaced with the word prosody contour which is a stylized prosody.

The syntax-driven prosody generation will be discussed in detail in the next chapter. For the remainder of this chapter, the implications of a direct derivation of the prosody from the syntax tree on the construction of the grammar will be discussed.

## 3.2 Minimalistic Grammar for Prosody

The general prosody generation method just introduced does not put any constraints on the kind of syntax tree used to describe the sentence. In principle, any grammar is suitable as long as it can be used to successfully parse the sentence. However, meaningful prosody can only be learned if the syntax tree is structured in a way that reflects the functional prosody of the sentence. The grammar of the PolySVOX system does not fulfill this requirement because it is mainly geared towards syntactic correctness. The remainder of this chapter discusses the modifications that were made in order to make the grammar prosody-aware. This section first gives a short introduction into the specific DCG variant used in SVOX and then describes the general rules that have guided the grammar development.

within their category the words can be replaced freely if semantics are not taken into account.

### 3.2.1 SVOX DCG Grammar

The SVOX parser is based on the definite-clause grammar (DCG) formalism, which provides a simple and efficient way of formulating and parsing grammars. A rule is written in the form of

$$\begin{array}{lcl} \texttt{HEAD}(a_1,\ldots,a_h) & \rightarrow & \texttt{BODY\_1}\ (a_{11},\ldots,a_{1n}) \\ & & \texttt{...} \\ & & \texttt{BODY\_N}\ (a_{m1},\ldots,a_{mn}) \\ & & \texttt{[:INV]} \\ & & \texttt{[}\textit{penalty}\texttt{]} \end{array}$$

which corresponds to conventional DCG rules with two extensions:

- `:INV` marks invisible rules. These rules are used during parsing in the usual way but their head node will be hidden in the final output tree. Thus, trees can be flattened using invisible rules, which in particular avoids strong right branching that is often found in syntax trees created with DCG grammars. For the prosody-centric parsing, it also allows to hide constituents that have a purely syntactic function, thus reducing the size of the output tree in general.

- Per-rule penalties are taken into account when disambiguating the parse results. SVOX uses a simple additive scheme to score the final parse trees, summing up the penalties of all constituents in the tree to obtain the final score of the tree. These penalties are mainly used to ensure that multi-lingual parsing does not yield unexpected results and to a lesser extent to give a penalty to unusual grammatical structures so that scoring ensures that more frequently used structures are preferred in the case of ambiguities. Penalties are manually assigned.

The grammar is split into a word and sentence grammar corresponding to the two parsing stages which will be explained in more detail in chapter 5. In this chapter, the main focus is on the sentence grammar because the parse tree from word parsing is not retained for further processing.

### 3.2.2 Modifications for Prosodic Coverage

The grammars of PolySVOX provide a solid and functional base for creating syntax trees particularly for the generation of prosody. We therefore decided to keep the DCG paradigm, even though other grammar types have been developed in the last years that might be able to express certain aspects of intonation structure better. However, the existing rules still were insufficient for the processing of audio books. The design of the grammar needs to fulfill two additional and essential requirements: it needs to be able to parse texts of novels and stories and it needs to carry the prosodic information discussed so far.

#### Extended Coverage for Prose

The existing SVOX grammar rules have been developed mainly for use with grammatically sound, isolated sentences. Fictional texts pose more of a challenge because a much wider range of syntactic constructs is used. For one part, incomplete sentences like ellipsis constructs or isolated phrases are relatively frequent. For another, sentences may become very complex spanning entire paragraphs and using grammatical constructs that are rarely found in other contexts. The SVOX parser contains a mechanism to cope with both constructs by creating artificial sentences: if a sentence cannot be parsed, the parser will compute the best sequence of partial parse results and unite them under a special sentence root. While such a construct was sufficient for rule-based prosody generation, the coverage of infrequent syntax structures is more important for a data-driven approach because they often indicate a special intention of the writer that requires special prosody. One goal of the grammar extension was therefore to increase coverage of the grammar. This includes adding a partial coverage of incomplete sentences.

#### Reflecting Prosodic Structures

The grammar should also reflect the functional prosody of the sentence. This means that the rules should explicitly include properties that can be derived from the text and may change the prosody rendering but have no direct impact on the syntax structure. In this, the syntax-driven prosody generation does not differ from other rule-based approaches. However, while standard rule-based approaches require to state *how* a syntactic construct modifies prosody, here it is only necessary to formulate rules for constructs *that are likely* to be important for prosody. The actual realization will then be learned. This task is much more simple and effectively only means increasing coverage of the grammar as well.

For hierarchic prosody models, tier systems were developed that are designed to follow the rhythmic and melodic structure of speech. The main layers found in all models are utterance, intonation phrase and word[3]. Some models add intermediate tiers to this like small phrases. Even though some authors insist that syntactic and prosodic structure have no direct correspondence [Bol72], the above tiers still are at least partially reflected in the syntax tree. The word tier is directly available in the leaf nodes of the tree. The phrase tier is more difficult to establish. [BF90] remarks that for phrasing mainly noun phrase (NP), prepositional phrase (PP) and adjective phrase (ADJP) are relevant to establish phrasing. In addition, verb groups provide the binding between these phrases. Finally, the utterance corresponds to the clause level.

A prosody-oriented syntax tree needs to reflect these three levels of constituents with equivalent prosody function: word, phrase and clause. In addition, the sentence level is of some importance because it allows to express relationships between clauses which are an important part of prosodic function. The second goal of the grammar extension is to restructure it with focus on these tiers.

### Design Principles

To fulfill the requirements just discussed, the grammar rules were revised according to the following principles:

**Simplification** Grammatical correctness does not need to be verified by the parser. It can be assumed to exist in the source or, where it does not exist, to be intentionally violated. Therefore, any constraints have been removed that only restrict grammaticality but have no impact on the correct structure of the parsing result.

[3] Prosodic models generally also define a syllable and segmental tier but these are covered by the segmental model and are not a subject for the grammar construction.

For example, German verbs, like in many other languages, require the use of a particular auxiliary verb, *haben* or *sein*. Wrong usage will cause a sentence to be ungrammatical but it does not produce ambiguities for the parser. Tracking of auxiliary verbs was therefore removed. This kind of simplification also greatly increased coverage for unusual grammatical constructs.

**Flattening** Many constituents in the tree have primarily a structural function. These have little meaning in a syntax-based learning approach. By hiding purely syntactic constituents and focusing instead on those representing the prosodic tiers discussed above, the complexity of the tree can be reduced and consequently also the complexity of the prosody model. An example is the flattening of clause structures as will be shown in figure 3.4 below in section 3.3.3.

**Prosodic Annotation** Constituent attributes were so far only used to express syntactic constraints. For the prosody-focused grammar, the use of attributes was extended to annotated the tree with functional prosody attributes as far as they can be inferred from syntax. The prosody model makes a locality assumption, as will be explained in more detail in the next chapter. It states that prosodic attributes are assumed to appear in the vicinity of constituents for which they are relevant. Therefore, the prosodic attributes are also carried upwards in the tree where necessary.

## 3.3 Grammar Coverage

In this final section, the coverage of the grammar will be discussed in detail according to the prosodic tiers introduced above. The focus is less on a description of a concrete implementation of grammar rules but more on establishing the general principles that guided the construction without reference to one language in particular[4]. The concrete realizations were guided by [CM06] for English and by [Dud09] for German.

[4] Examples in this section will be accordingly given in a generalized language-independent notation. In the actual grammars the rules are always duplicated and adapted to the specific peculiarities of each language.

| **Noun** | **Verb** | **Adjective** | **Adverb** |
|---|---|---|---|
| common noun | full verb | grade | adjectival |
| proper noun | auxiliary verb | | defining |
| measure word | modal verb | | temporal |
| number | | | causal |
| title | | | negative |

Table 3.1: *Content word categories and their sub-groups*

### 3.3.1 Word Level

The word tier is made up of the leafs of the syntax tree. Even though the internal structure of words is parsed, it is not retained for the prosody generation step. Thus there are only two kinds of information available at this level: the part-of-speech as it is encoded in the constituent and word-level attributes. In terms of their interest for prosody, words can be roughly divided into content words and functional words.

For the open word categories noun, verb, adjective and adverb, the most interesting property is their function. They have been each divided into subgroups as listed in table 3.1 to improve syntax parsing. These groups are also relevant for prosodic rendering. The prosody of verbs and nouns is different for the different types. The subgroups of the adverb class help to determine the function of phrases and clauses. Proper names also contain an attribute to further categorize the following types: given names, surnames, geographic names and spelled out words. Beyond that little information can be extracted without a semantic analysis.

Of greater interest for a prosodic analysis are pronouns and other function words. Although they are normally unaccented unless particularly emphasized, they function as markers to indicate changes in neighboring words or enclosing phrases. The following gives an overview over the categories that were taken into account for prosody together with the relevant attributes.

**Determiners and pronouns** Defined in this category are: articles and determiners and personal, demonstrative and indefinite pronoun. They also carry an additional type attribute to distinguish between definite, indefinite and negative connotation. For determiners this attribute is carried over to also define the connotation of the noun phrase.

**Conjunctions** In absence of any semantic analysis, the conjunction word is the main indicator for the type of coordination that connects words and phrases. The grouping here follows the main types defined in [MR03]: addition, comparison, time and consequence.

**Prepositions** can be grouped into frequently used primary prepositions and less frequently used secondary ones. The latter can be further split into simple and complex prepositions. Complex ones may be composed of multiple words but may also be single-word proposition derived from more complex collocations (compare German *mithilfe*).

**Particles** may draw the focus to a part of the sentence. The functional groups follow the categories used in [Dud09]: grading, negative, modal, focus, structural, answering and interjection.

### 3.3.2 Phrase Level

The basic syntactic phrase types are noun phrase, prepositional phrase, adjective phrase and adverbial phrase. Syntactic phrases are the most important element in the syntax tree because they are closest correlated to intonation phrases. However, there is no direct correspondence between the two. Syntactic phrase types can become very complex, including being nested. In the latter case, they may span multiple intonation phrases. Conversely, an intonation phrase may be made up of multiple short syntactic phrases. However, the grammar can be structured in a way that the boundary between intonation phrases always falls on the boundary of a syntactic phrase (or sub-phrase in the nested case).

#### Noun Phrases

The structure of Noun phrases (NPs) follows the general composition principle as shown in figure 3.2. The phrase structure remains flat. Only the nucleus is made explicitly visible for two reasons: first, the

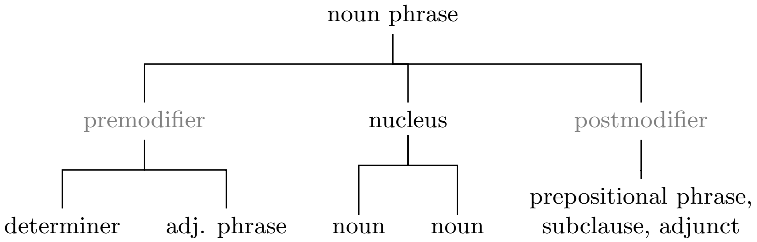


Figure 3.2: *General structure of the noun phrase. Gray nodes represent invisible parts of the syntax tree. All parts (including the nucleus) are optional.*

| type attribute | determiner | nucleus |
|---|---|---|
| `def` | definite | noun |
| `indef` | indefinite | noun |
| `neg` | negative | noun |
| `name` | *any* | proper noun |
| `pers` | *any* | personal pronoun |
| `pron` | *any* | other pronoun |
| `num` | *any* | numeral or year |

Table 3.2: *Sub-types attributes for noun phrases according to the content of the determiner in the premodifier and the nucleus of the noun phrase.*

nucleus serves as a structural marker separating premodifier and postmodifier. Second, the word type of the nucleus more closely defines the type of the noun phrase. On one side, there are proper noun phrases with a noun in the nucleus. On the other side there are noun phrases consisting of pronouns and determiners only which often receive a very different prosody, sometimes even being unaccented. Accordingly, the type of noun phrase is further refined in an additional attribute as shown in table 3.2.

| Construct | Example sentence |
|---|---|
| double preposition | *Sie parkte* ***unweit vom Schloss****.* |
| coordinated preposition | *Er suchte die Nadel* ***auf und unter dem Tisch****.* |
| contrasted preposition | *Er suchte die Nadel* ***sowohl auf als auch unter dem Tisch****.* |
| noun ellipsis | *Vorsicht ist angebracht* ***bei Haus- und bei Gartenarbeit****.* |

Table 3.3: *Special cases of coordination in prepositional phrases covered by the English and German grammars. Examples are provided for German only.*

### Prepositional Phrases

In its standard form, prepositional phrases (PPs) consist of a preposition in preposition or postposition and a noun or adverbial phrase, e.g.

$$\begin{array}{rcl}
\texttt{PP(}\mathit{pre},\ \mathit{?nptype}\texttt{)} & \rightarrow & \texttt{PREP(}\mathit{pre}\texttt{) PP\_TAIL(}\mathit{?nptype}\texttt{)} \\
\texttt{PP(}\mathit{post},\ \mathit{?nptype}\texttt{)} & \rightarrow & \texttt{PP\_TAIL(}\mathit{?nptype}\texttt{) PREP(}\mathit{post}\texttt{)} \\
\texttt{PP\_TAIL(}\mathit{?nptype}\texttt{)} & \rightarrow & \texttt{NP(}\ldots,\ \mathit{?nptype}\texttt{)} \\
\texttt{PP\_TAIL(}\mathit{adv}\texttt{)} & \rightarrow & \texttt{ADVP ()}
\end{array}$$

As can be seen, prepositional phrases also carry over the noun phrase type as described in table 3.2.

In addition to this standard form, a number of special cases of coordination have been implemented that are not covered by the general coordination construct explained below. They are listed in table 3.3 together with German example sentences.

### Other Phrase Types

Next to the primary phrase types, the following secondary phrase types have been added as their own type to the grammar: comparison phrases and time and date expressions. The latter also includes more general temporal expressions like *some time ago.*

### Coordination

A particularity that appears frequently at phrase level is the coordination of word groups. These pose a problem for the construction of the grammar because almost any part of a phrase may be coordinated as well as complete phrases leading to an explosion of coordination rules.

To allow coordination on arbitrary levels, a special set of meta-rules was introduced with a general coordinated term COORDL and the general coordination constituent COORDR. The different types of coordinations were then modeled using these two constituents. The type of coordination was added as an attribute to the coordinator. For example, the rule for coordination with opposition has a rule like this:

```
COORDR(op,?C,?P1,?P2,?P3)   →   COORDL(?C,?P1,?P2,?P3)
                                CONJCOORD(op)
                                COORDL(?C,?P1,?P2,?P3)
                                :INV
```

where $?P1$ to $?P3$ are arbitrary attributes that need to be matched in the coordinated terms and $?C$ is an identifier for the constituent to match. The rule itself remains hidden.

In the phrase grammar, the general coordination is mapped to the specific constituents to be coordinated, for example:

```
      NP(?C,pl,pers3,coord)   →   COORDR(?TP, np,?C,?,pers3)
COORDL(np,?C,?FUNC,?PERS)     →   NP(?C,?NUM,?PERS,?FUNC)
```

When full phrases appear in coordination, their prosody is different enough from their uncoordinated counterparts that they should be generated from a different prosody function. They therefore receive a different constituent. The full set of rules for NP coordination becomes thus:

```
   NP(?C,pl,pers3,coord)   →   NP_SIMPLE(?C,?N,?P,?F)
   NP(?C,pl,pers3,coord)   →   COORDR(?,np,?C,?F,?)
     COORDL(np,?C,?F,?)    →   NP_COORD(?C,?N,?P,?F) :INV
  NP_COORD(?C,?N,?P,?F)    →   NP_SIMPLE(?C,?N,?P,?F)
 NP_SIMPLE(?C,?N,?P,?F)    →   ...:INV
```

The result is again a flat tree structure, as can be seen for the exemplary syntax tree for the coordinated phrase *the good, the bad and the ugly* in figure 3.3.

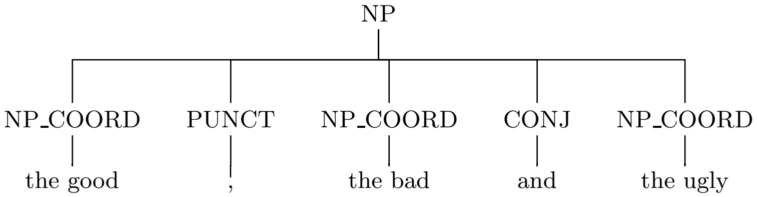


Figure 3.3: *Example of coordinated syntax structure for the phrase* the good, the bad and the ugly.

### 3.3.3 Clause Level

Clauses connect the phrases with a verbal construct, thus defining the relationship between them. Classic grammars tend to use deeply nested structures to define these relationships. Such hierarchies are disadvantageous for our case for three reasons. First, the nesting causes locality to be lost. The structure of the clause can only be determined by descending the entire tree. Second, to obtain a correct analysis, detailed knowledge of verb valency and the grammaticality of phrase compositions is required. This goes against the goal of simplicity as defined at the beginning of this section. Finally, it has been observed many times (e.g. [BF90]) that the phrase structure within a clause does not necessarily follow the right-bracketed structure of syntax. Phrasal balancing and information structure play a more important role in determining the prosodic structure.

Clauses are therefore maintained in a flat single-layer structure where all syntactic parsing details are hidden. Verb balancing and adjacency can then be described through simple features of the clause node, in particular number, type and ordering of its children. Figure 3.4 shows an example of the flatting process for the German clause structure *Zweitsatz.*

The grammar distinguishes three clause types: main clause, subordinate clause and infinite clause. Subordinate clauses are further split into nominal, subjunctive and relative clauses. Main clauses may be divided further according to the word order in the specific language. This classification also reflects different prosodic realizations.

The second important factor for prosody at this level are functional

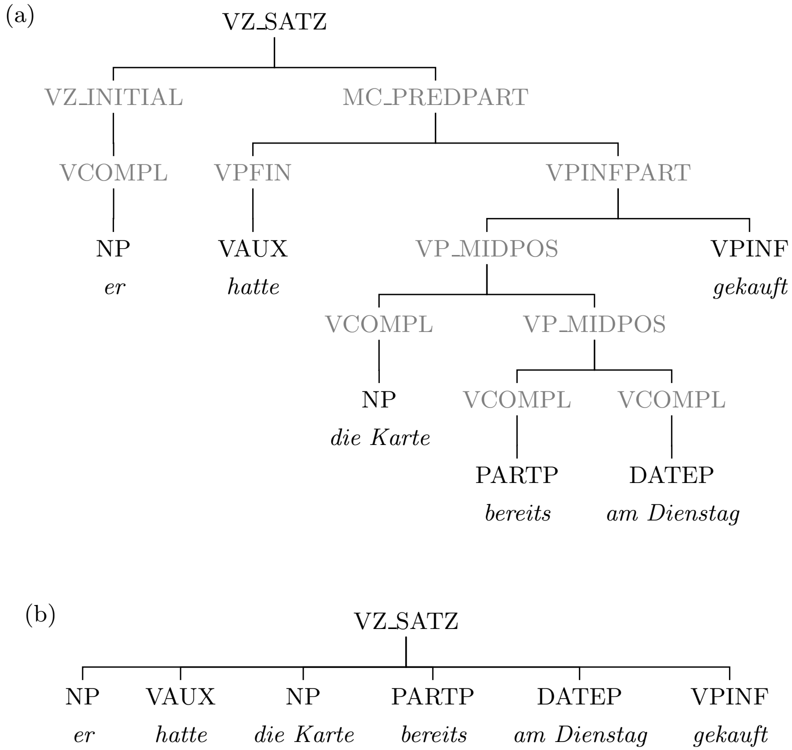


Figure 3.4: *Example of clause flattening for the German clause "er hatte die Karte bereits am Dienstag gekauft" (he had already bought the ticket on Tuesday): the complete parse tree (a) with invisible constituents (in gray) includes Vorfeld (VZ_INITIAL), Satzklammer (VPFIN, VPINF) and Mittelfeld (MIDPOS) structures as defined in [Dud09]. After hiding invisible constituents, only a flat clause structure (b) is returned.*

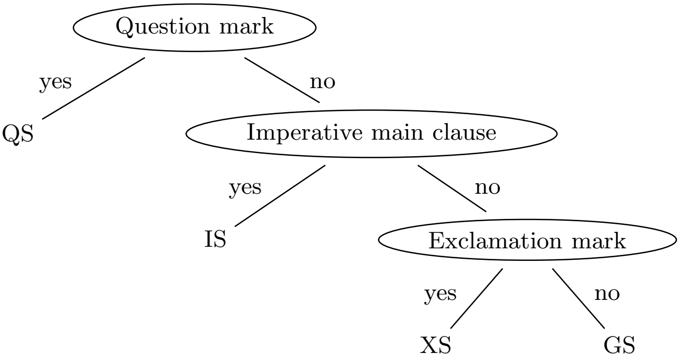


Figure 3.5: *Assignment of the primary sentence function according to type of main clause and punctuation. The functions are: question (QS), command (IS), exclamation (XS) and statement (GS).*

properties of the phrases—as discussed in the previous paragraph—and the verb. In particular, the function of the subject (relative, interrogative or declarative) and the mood of the verb are of interest.

Missing in this list are adverb phrases. These can give important additional hints about the role of the clause within the sentences and of the sentences within the discourse. However, without a semantic analysis, it is in general difficult to determine the context to which adverb phrases relate. They are therefore not considered beyond their function as a child node of the clause.

### 3.3.4 Sentence Level

The most important prosodic attribute of the sentence is its function which can be determined syntactically from the clause structure and punctuation. Figure 3.5 illustrates the assignment process. The main functions are: statement, questions, exclamation and command. For coordinated sentences, the final main clause determines the function.

The main clause of the sentence may also be accompanied by sentence adjuncts. They may appear at the beginning or end of the sentence. If a sentence consists of multiple coordinated main clauses, the

| adjunct type | structures covered |
|---|---|
| `inj` | particle, interjection, adjective<br>***Oh***, *I forgot my keys!* |
| `addr` | proper noun phrase<br>*You are next in line,* ***Mr Smith***. |
| `phrase` | noun phrase, prepositional phrase, comparison phrase, time phrase<br>***In the morning,*** *the car had disappeared.* |
| `clause` | infinitive and subordinate clauses<br>*He loves going into town on Friday nights,* ***especially now that the Christmas lights are up***. |

Table 3.4: *Implemented sentence adjunct types, the grammatical structures they represent and an example use for English.*

adjuncts may also appear between the parts of the coordination. Adjuncts are particularly frequent in written texts. Like noun phrases, they can have a very diverse structure and complexity. Therefore, these adjuncts are implemented as a special clause type that exports its content type and function via attributes. The adjunct types are listed in table 3.4. When appearing in postposition, it is not always apparent from the syntax if a sentence part is an adjunct or part of the main clause. For clause adjuncts, the adjunct position is preferred in these cases. All other phrase types are penalized in adjunct position.

Finally, tag sentences were added to the grammar. These include short subject-verb sentences ("You go!") and exclamations and question that consists only of a single word or a short phrase ("What a mess!"). Syntactically, these sentences could be handled correctly as artificial sentences. However, these short sentences and exclamations have often a distinct prosody that is different from that of full sentences. A special category will simplify learning of this prosody and avoid contamination of the prosody models for standard sentences.

## 3.4 Discussion

In this chapter, we have introduced the general concept of a statistical syntax-driven prosody generation and outlined the elements necessary in the grammar so that the syntax tree is geared towards a prosodic analysis.

The implemented grammar results in syntax trees that are already strongly related to conventionally used prosodic layers while still preserving the syntax structure. Each layer preserves the locally important information so that a localized analysis of the tree for prosodic properties becomes possible.

The statistical generation method aims to leverage more directly the relationship between syntax structure and prosody realization. Using a data-driven approach, the relation is learned directly from examples of natural speech without the need to explicitly annotate abstract prosodic events. Dropping the constraints of prosody abstraction and narrow feature contexts means that such a system has much more degrees of freedom than existing systems. To still be able to train such a system in a meaningful way, the design has to concentrate in particular on the following points:

**Syntax-to-prosody mapping** While the model of base prosody composition simplifies the generation process, it makes the training of the model more complex because the prosody of the training material needs to be decomposed into its base components. The large feature space means that the training has to be able to cope with sparsely distributed features. And finally, the observable prosody is not only a product of the syntax structure of the sentence but is also influenced by other factors like intent and emotions. The result is that training material for prosody is always noisy.

**Training material** The large feature space entails that the training data needs to cover a large number of variants of syntax structures which should be spoken in their natural context. Linguistic prosodic corpora of isolated sentences fulfill neither of these criteria. A much better source is training material from the same domain as intended for synthesis as they have the same syntactic and prosodic style. This material is in general not explicitly recorded for speech processing and needs to be preprocessed and annotated accordingly. An automation of the process can greatly reduce development time for new prosody models and facilitate porting to new languages and domains.

Each of these topics will be discussed in detail in the remainder of this thesis starting with an in-depth discussion of the prosody models in the next chapter.

# Chapter 4

# Syntax-driven Prosody Generation

Using the grammar developed in the last chapter, a compact syntax tree is generated that through its structure and through its constituent attributes encodes a part of functional prosody. This chapter will now discuss in detail the process of computing the concrete prosody from the syntax tree. We will first introduce the general idea in more detail and give a formal definition of the problem. Then the most important elements will be discussed: the statistical models for computing the prosody, the features used by these models and the elements of the word prosody contour that functions as an interface between tree-derived and segmental models. The chapter closes with some remarks on the training of the model.

## 4.1 Concept

The goal of the syntax-driven prosody generation is to derive concrete prosody from a syntax tree by means of statistical models. Essentially that means to find a function that maps a hierarchical structure, the syntax tree, onto a linear structure, the speech signal. This is not straightforward because syntax trees can become arbitrarily complex, just as the parts of natural sentences can be deeply nested or coordinated. Statistical models require a feature vector of fixed size as input. Syntax trees, however, are not necessarily balanced and vary in depth, making it difficult to define a fixed set of features. The common solution for this problem in statistical prosody generation systems is to focus on selected parts of the structure that are particularly interesting for prosody and derive a fixed set of features from those. This approach assumes that we know in advance which properties of the tree are most relevant for the realization of concrete prosody. The goal of the system presented here is to no longer make this assumption. Instead the model should learn which syntax structures are relevant at the same time as learning how they influence concrete prosody. To be able to handle the complexity of the syntax tree without preliminarily discarding potential features for prosody generation, we follow a decomposition principle where the syntax tree is broken down into a collection of its nodes and each of them is responsible to produce part of the prosody. Thus, we assume that the concrete prosody can be decomposed into a set of *base prosody contours*, each of them representing a specific aspect of the functional prosody that is encoded in the syntax structure of the tree. The base prosody contours can be defined through the following properties:

- Each base prosody contour is assigned to exactly one node in the syntax tree, the *base node*.
- The context of the node within the tree determines the realization of the base prosody contour.
- The base prosody contour is confined to the part of the sentence that is covered by the sub-tree of which its base node is the root.
- The syntactic influence on the concrete prosody of a sentence is equal to the composition of these base prosody contours.

Relating base prosody contours directly to tree nodes means that the syntax tree itself can be considered a prosody generator. Each node is assigned a *mutator function* that computes the base prosody contour with respect to the node context. The final concrete prosody is then the result of the concatenated application of all mutator functions in the tree. As the context of each node is well defined and restricted by the grammar, it can be described with a fixed set of features, so that each mutator function can be represented with a statistical model allowing a data-driven learning approach for the complete prosody model.

The mutator functions only take into account the node context within the syntax tree. Because the syntax tree only describes the sentence structure with words as the basic elements, the model does not include syllable or phone properties of the words. These *segmental features* have an important influence on the concrete prosody but are largely independent of the functional property encoded in the syntax structure. Hierarchic models generally consider the segmental level as an additional first-class layer that is combined with the other layers. We follow the same approach here. However, instead of computing a separate base prosody contour from the segmental features and combining them with the outputs of the mutator functions, the combined output of the mutator functions is considered the input to another statistical model that combines them with the segmental features and outputs the final concrete prosody. Such a model allows a non-linear transformation of the word prosody contour and as such can compensate for differences between words with respect to their length or the placement of lexical stress. Figure 4.1 illustrates this two-step process of prosody generation.

In the first step, a tree of mutator functions is created according to the syntax structure of the sentence. This is the *tree-derived model* of the sentence. When applied it results in a prosody at word level which shall be referred to as *word prosody contour*. Ideally this would be the stylized prosody of the word, that is the concrete prosody with all segmental influences removed. We will see later that a more crude stylization needs to be used. However, it is important to note that the word prosody contour can be directly computed from the speech signal and as such is not an abstract prosody as found in other prosody models including PolySVOX.

The second step consists of the *segmental model*, which is responsible for computing the concrete prosody parameters. This is also realized as a statistical model whose input vector is composed of the word prosody contour and the segmental features. The segmental features are extracted from the phonetic transcription that has been added to the words in the syntax tree in previous computation steps.

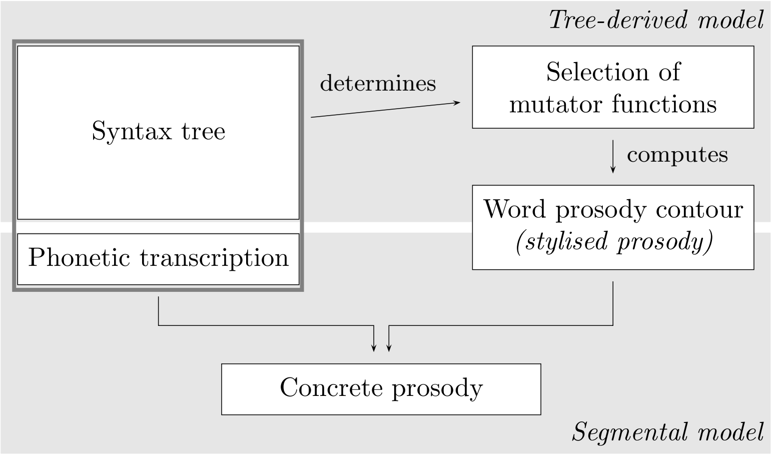


Figure 4.1: *Two-step process of determining prosody. The tree-derived model determines a stylized prosody for the word from the information contained in the syntax tree, and the segmental model produces the concrete prosody from that and features from the phonetic transcription.*

## 4.2 Formal Definition

The purpose of the prosody generation module is to compute the concrete prosody $\Pi$ of a sentence from its syntax tree $\tau$.

The syntax tree is an ordered tree and can be described in a simplified way by the tuple $\tau = (\mathbf{N}, \rho)$ where $\mathbf{N}$ is the set of nodes and $\rho$ is a function that describes the parent-child relationship between these nodes. This function is defined as

$$\rho : \mathbf{N} \times \mathbf{N} \rightarrow \mathbb{N}$$

where $\rho(N_p, N_c) = i$ for $i > 0$ denotes that $N_c$ is the $i$-th child node of node $N_p$ and $\rho(N_p, N_c) = 0$ means that $N_p$ and $N_c$ have no parent-child relationship. Nodes are tuples $N = (C, \mathbf{A})$ of the constituent $C_N$ of the node and a set of attributes $\mathbf{A}_N$. The tree has exactly one root node $N_0 \in \mathbf{N}$ with

$$\forall N \in \mathbf{N} \quad : \quad \rho(N, N_0) = 0$$

and an ordered set of leaf nodes $\mathbf{L} \subset \mathbf{N}$ with

$$\forall L \in \mathbf{L}, N \in \mathbf{N} \quad : \quad \rho(L, N) = 0.$$

Each leaf $L_i$ represents a word of the sentence or a punctuation mark. Words are assigned a phonetic transcription $T_i$. If the leaf contains a punctuation mark, then the phonetic transcription remains empty.

We further need to introduce the notion of descendants. A node $N_d$ is a direct descendant of a node $N_r$ when it is part of the subtree headed by $N_r$ or in other words, there exists a direct path from $N_r$ to $N_d$. The set of descendant tuples $D$ is formally defined as

$$\begin{aligned} (N_r, N_d) \in \mathbf{D} \iff \quad & \exists\, M_1, \ldots, M_m \in \mathbf{N} : \\ & M_1 = N_r, M_d = M_m, \\ & \forall 1 \leq i < m : \rho(M_i, M_{i+1}) > 0 \end{aligned}$$

### Two-Stage Prosody Synthesis

The concrete prosody of the sentence is described by a sequence $\Pi$ of prosody vectors. These are the parameters passed on to the speech signal generation. Without loss of generality, the concrete prosody of each word in the sentence shall be considered separately. For each leaf $L_i \in \mathbf{L}$ there is a well defined sub-sequence $\Pi_i$ of $\Pi$ that covers the word described by $L_i$. Thus, the prosody of a sentence is generated by computing the concrete prosody of each word and concatenating the results.

As the prosody generation is a two-stage process, the word leafs are also assigned a stylized prosody. For each $L_i$ there is a vector $P_i$ that describes the word prosody contour for the attached word. Punctuation leafs have no corresponding word prosody contour, they are skipped during the generation process (although they are still taken into account in the context of neighboring nodes).

The word prosody contour vector can be computed directly from the concrete prosody. Given a prosody vector $\Pi$ of the natural prosody of a word, there exists a function

$$P = W(\Pi)$$

that determines the stylized prosody. This function is only required to compute the target vectors during training. It allows to train the tree-derived model separately from the segmental model.

For the application of prosody, the inverse function is needed and corresponds to the application of the segmental model. The computation of the final concrete prosody of the word is straightforward and defined as

$$\Pi_L = S(P_L, T) \tag{4.1}$$

where $S$ is the segmental prosody generation function and $T$ the set of segmental features. Note that the segmental features are not confined to the word to be computed but also cover neighboring words.

### Mutator Functions

Now the tree-derived model remains to be formally defined. Following the super-positional principle of the hierarchic methods as discussed in 2.5.4, we assume that the word prosody contour can be decomposed into base prosody contours originating from each node in the syntax tree. The generation process consequently needs to produce these base contours and combine them into the final concrete prosody. To that end, a mutator function is defined that describes modifications to the word prosody contour. Given a syntax tree $\tau$ and an intermediate word prosody contour $P_{in}$ of a word, the mutator function $M$ returns a new word prosody contour $P_{out}$ for the word:

$$P_{out} = M_N(L, \tau, P_{in})$$

where $N$ is the *base node*, the node in the tree, which is the origin of the mutation, and $L$ the *target node*, the leaf node where the prosody modification is applied. In this definition, the mutation depends on the complete syntax tree which is not a feasible solution. Two locality assumptions are added, *feature locality* and *scope locality.*

Feature locality means that only properties of nodes in the immediate neighborhood of the base node and the target node are assumed to be relevant for the computation of the output prosody. Thus the mutator function can be reduced to

$$P_{out} = M_N(F_\tau(N, L), P_{in})$$

where $F_\tau(N, L)$ denotes the vector of locally relevant features for base node $N$ and target node $L$ in tree $\tau$. The exact content of this vector will be discussed in more detail in section 4.5.1.

Scope locality defines that mutations originating from a node in the tree are only applicable to leaf nodes that are within the sub-tree of the nodes and are thus direct descendants. Conversely, they do not change the prosody of leaves that are not direct descendants, formally

$$\forall N \in \mathbf{N}, L \in \mathbf{L} : (N, L) \notin \mathbf{D} \implies M_N(F_\tau(N, L), P) = P$$

Thus, for the concrete prosody of a leaf node it follows that it is only modified through prosody mutations originating from nodes that are on the direct path between the root node $N_0$ and the leaf node. Let $N_0, N_1, \ldots, N_p, L$ be the sequence of nodes on this path, that is $\forall\, 0 \leq k < p : (N_k, N_{k+1}) \in \mathbf{D}$ and $(N_p, L) \in \mathbf{D}$. Then the computation of the prosody for leaf node $L$ can be defined recursively as

$$P_L^k = M_{N_{k-1}}(F_\tau(N_{k-1}, L), P_L^{k-1}) \qquad 0 < k \leq p \tag{4.2}$$

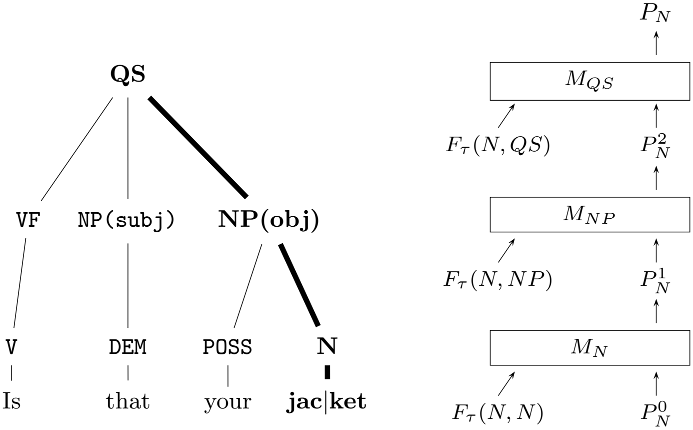


Figure 4.2: *Bottom-up application of mutators for the word 'jacket'. $P_N^0$ is the initial neutral prosody contour, $P_N$ the final contour and $F_\mathcal{T}(N, C)$ are the input features of the mutator for constituent $C$ in the tree.*

where $P_L^0$ is an initial neutral word prosody contour and the final word prosody contour is then

$$P_L = M_L(F_\mathcal{T}(L, L), P_L^p) \tag{4.3}$$

for a top-down application of the mutator functions. The mutator functions can also be applied bottom-up, in which case the sequence of nodes will be $L, N_p, \ldots, N_1, N_0$. Figure 4.2 illustrates the concept of the recursive computation of the word-prosody contour for the bottom-up application of mutators.

The order of application of the mutator functions is important because of the recursive nature of this definition. The scope of the mutator functions decreases with increasing distance from the root node of the tree. Thus, a top-down application of mutators implies that the generation starts with a general sentence contour which is subsequently refined on a more and more local scope. While this method is linguistically more plausible, there is an opposite effect of importance that needs to be taken into account. We have observed that the influence of a mutator on the final word prosody contour is the stronger the closer it is to the leaf node which would suggest that bottom-up application is better suited. An experimental evaluation of both approaches will follow later in this chapter in section 4.6.

The next section shortly discusses the realization of the segmental models. Then the remainder of this chapter will focus on the tree-derived model. First the realization of the mutator function will be detailed including a discussion of the input features followed by an evaluation of the realization of the word prosody contour and finally the training of the statistical models will be outlined.

## 4.3 Segmental Model

The segmental model takes the word prosody contour and the segmental features as input parameters and produces the final concrete prosody at phone level. The process is very similar to the Prosody Generation stage in PolySVOX with the exception that instead of the abstract prosody in the phonological transcription the word prosody contour is used. As artificial neural networks (ANNs) have proved to produce close to natural prosody, the same models have been used in this segmental stage.

### 4.3.1 Output Parameters

The speech signal generation of SVOX expects two prosodic parameters for each phone: its duration and a $k$-point $F_0$ contour, where usually $k = 5$. While the segmental model computes durations directly for each phone, $F_0$ is computed syllable-wise. The $F_0$ contour for the phone is produced from this by piece-wise linear interpolation of the syllable contour.

Both, $F_0$ and duration are normalized to compensate for physical voice characteristics. $F_0$ values are converted into Mel frequencies and then normalized with respect to the voice range of the speaker. For the training voices, the $F_0$ range is computed directly from the training material. For synthesis it can be chosen freely for each voice.

The phone duration is normalized with respect to the individual durations of each phoneme in order to take into account the specific length and elasticity of each phoneme. This is similar to what was proposed in [WSHOP92]. The duration distributions are computed from the training material at the beginning of the prosody learning process. The normalization is realized as a piece-wise linear interpolation between the 1, 50 and 99 percentiles of the length distribution of each phoneme. These computed normalization parameters are retained as part of the prosody model and reused during synthesis.

### 4.3.2 Prosody Generation

The segmental model is realized with two independent ANNs. The duration ANN produces one value per phone, the $F_0$ model 5 support points per syllable: beginning of syllable, end of onset, center of nucleus, beginning of coda and end of syllable. The selection and context for the segmental features can be much smaller than what was proposed in [Rom09a] for PolySVOX because the word prosody contour already encodes the abstract prosody parameters in a much more compact form. The selection therefore follows largely what was proposed for classic SVOX. A detailed description of the configuration and the segmental features of the two ANNs can be found in appendix A.

## 4.4 Word Prosody Contours

The word prosody contour is the interface between the tree-derived model and the segmental model. As such it should represent a stylization of the prosody of the complete word with the segmental influence removed. There is no accepted definition of what constitutes the segmental part of prosody and, consequentially, there exists no straightforward way to separate super-segmental influences from the concrete prosody. Instead, a parametrization of the concrete prosody is used for the word prosody contour. We assume here that each prosody parameter contributes an independent dimension to the word prosody contour, so that the definition of these parameterizations can be defined separately. In this section, we investigate the goodness of different parameterizations for each of the following prosody parameters in detail: $F_0$, duration and presence of pauses. Intensity has been omitted because the SVOX signal generation is not able to model intensity. The general principles, however, can be extended to intensity as well once an appropriate signal generation method has been implemented.

Stylizing prosody at word level may not be obvious as it is well known that words are not a unit that is distinguishable in the speech signal. In [HP12], we have described a similar approach were the tree-derived model was producing directly stylistic prosody contours on syllable level. However, syllables show large differences in the realization of their concrete prosody between stressed and unstressed syllables. Because stress is related much stronger to the lexical content of the word than to the syntax structure of the sentence, this difference cannot be modeled in a meaningful way in the tree-derived model. Word prosody contours, however, are able to express better the general strength of accentuation of the word independently of its internal stress pattern. Syllable properties and lexical stress are then included as features of the segmental model, where the difference between stressed and unstressed syllables can be learned more directly.

### Evaluation Method

Before discussing the different prosody parameters in detail, this section introduces the evaluation method used. The objective of the evaluation of the different parameterizations is to understand how well they are suited as an intermediate representation between tree-derived and segmental model. An appropriate parametrization must be general enough to be free of any segmental influence, so that it can be derived from the syntactic features. It must also be close enough to the original prosody, so that the concrete prosody can be reconstructed from segmental features and the word prosody contour.

The suitability of the word prosody contour parameters for the segmental model can be evaluated directly. First, the *expected contour parameters* need to be computed from the natural prosody of the training sentences. Then a segmental model is trained on these expected contour parameters and the segmental features. The quality can then be evaluated by comparing the mean-square error (MSE) between natural and synthetic concrete prosody for each prosodic parameter. To be able to compare how much of the concrete prosody is derived from the word prosody contour parameters, an additional ANN is trained with segmental features as input only, later referred to as *baseline ANN*.

The performance of the tree-derived model cannot be tested directly because the different word prosody contour parameters have different distributions so that MSEs are not directly comparable. Instead, we compare them, similar to the evaluation of the segmental model, through the synthesized concrete prosody as follows: the synthesized prosody from the segmental model trained above is used as a reference by computing the synthesized prosody with the expected contour parameters. Then a tree-derived model is trained using the expected contour parameters as target output. With this model *computed contour parameters* are produced. With those a second synthesized prosody is produced using the same segmental model trained with the expected contour parameters. The MSE difference between the synthesized prosody with the expected and computed contour parameters is used as an indicator of how much of the word prosody contour parameters could be learned by the tree-derived model. This will be referred to as the *combined model*.

All MSE values are given for the normalized $F_0$ and duration values as described in section 4.3.1 because they compensate for the individual $F_0$ ranges and speaking rate of each speaker. This makes the results more comparable between voices. However, it does not account for individual differences in the variability of $F_0$ and duration. The MSE values for speakers with a more flat and neutral prosody are always lower than for speakers that vary their speech more.

An objective evaluation is sufficient at this point as the main interest is to understand how close the synthesized prosody is to the natural one. But it should be noted that an evaluation based on MSE of prosodic parameters does not necessarily reflect the subjective quality of the resulting synthesized prosody. However, subjective evaluations of prosody are known to be difficult and result in large disagreement between subjects. As the difference between the different parameterizations is relatively small, a subjective evaluation would need to be done on large amounts of data to be meaningful.

The training material in this chapter is comprised of two small speech corpora in German and English. They have been recorded specifically for prosody training by two different professional female speakers. The German corpus consists of 1500 isolated sentences and the English one of 1000 sentences. Both corpora contain mainly declarative sentences but they have been recorded in a lively speaking style and there are many syntactic constructs that are interesting for prosody production like enumeration, negations and contrasts. Tests were repeated 10 times, where each time 100 sentences were used as test sentences and the remaining material was used for training. The input features used for the tree-derived model will be discussed in detail in section 4.5.1 below, the segmental models are described in appendix A.

### 4.4.1 Fundamental Frequency

The stylization of $F_0$ has been well researched for syllables because a parametrization of the continuous $F_0$ curve is required for any prosody generation system. The RFC model [Tay94] describes the abstracted $F_0$ in terms of a rise-fall contour. It was later simplified into the Tilt model [Tay00] which describes $F_0$ only in terms of either rise or fall events. Thus it can be restricted to only three parameters: duration, amplitude and tilt. The MOMEL algorithm [HCB$^+$93] defines the $F_0$ contour in terms of quadratic spline functions. Its interpolation points are normally manually defined. To obtain a representation that can be automatically derived, INTSINT [HC98] was developed. It uses a categorical representation of intonation patterns, defining the tone globally with respect to the speaker's voice range (top, mid, bottom) and locally with respect to the preceding tone (high, upstepped, same, downstepped, lower). Finally, the model of quantitative target approximation [POXT09] is a model motivated by physiological constraints and describes $F_0$ in terms of target high, target slope and rate of target approximation. PolySVOX defines the syllable $F_0$ in terms of well defined support points (onset, nucleus, coda). The final $F_0$ contour is reconstructed via linear interpolation between these points. The number of support points per syllable can be freely chosen and constitutes a trade-off between fine-grained modeling and the size of the ANN model that produces it.

These syllable stylizations have in common that they aim to allow a reconstruction of $F_0$ that is as close as possible to the natural concrete prosody. The parameters for the word prosody contour, however, require a certain level of abstraction of the $F_0$ contour of the word that discards any segmental influence. While there is a general agreement

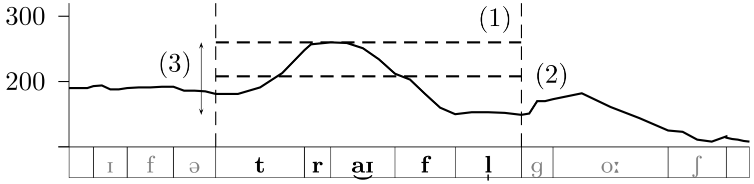


Figure 4.3: *Computation of parameters for extrema $F_0$ stylization for the word* trifle *in the example phrase "if a trifle gauche": (1) maximum $F_0$, (2) median $F_0$ and (3) maximum extent.*

in the literature that segmental feature have a significant effect on the $F_0$ contour ([Sil87]), the exact extent of this influence is unknown. In the following, three different stylizations are evaluated that model the word $F_0$ with an increasing level of detail:

**Extrema** This most coarse-grained stylization only defines constraints for the average and extremes of $F_0$ realization for the word. The following values are computed over the $F_0$ curve within the word as illustrated in figure 4.3: (1) maximum $F_0$, (2) median $F_0$ and (3) difference between minimum and maximum $F_0$ (movement).

**Local gradient** This stylization aims to also describe the general direction of $F_0$ movement within the word. The $F_0$ contour is approximated by a linear function using a simple least-square regression and then this approximation is described by the parameters shown in figure 4.4: (1) $F_0$ of the word center (absolute positioning of $F_0$), (2) $F_0$ at the beginning of the word (indirectly representing the direction of $F_0$ movement) and the MSE of the approximation (flatness of $F_0$ curve).

**Support points** The final stylization models the contour directly by using $F_0$ values on well defined points of the word. The position of the support points needs to be related to the content of the word to ensure they are duration-independent. The center points of three syllables are used as shown in figure 4.5: initial and final syllable and the syllable that carries the main stress. For unstressed words, the center syllable, or, in case of an even number

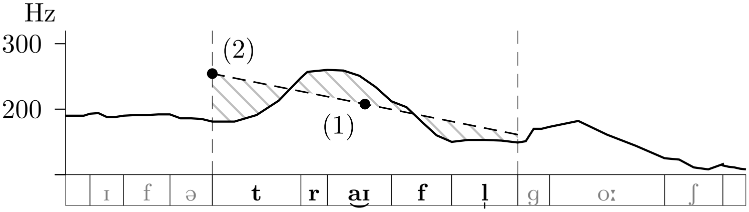


Figure 4.4: *Computation of the local gradient parameters for the word* trifle *in the example phrase "if a trifle gauche" consisting of $F_0$ at the word center (1), $F_0$ at the beginning of the word (2) and deviation from the regression line (shaded area).*

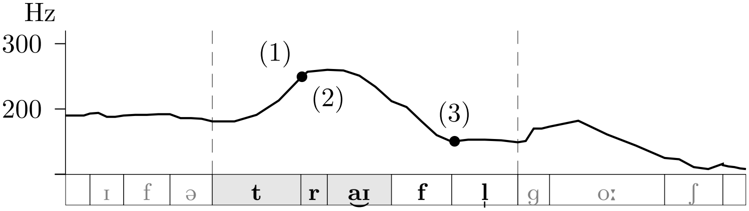


Figure 4.5: *$F_0$ points used in the support point stylization for the word* trifle *(with main stress on syllable* tri *in gray) in the example phrase "if a trifle gauche". The center of the first syllable (1) and stressed syllable (2) fall together. (3) represents the center of the final syllable.*

of syllables, the syllable before the center is used. These three parameters are also well-defined for mono-syllabic words although they will all fall on the same syllable.

The evaluation results for these three stylizations are listed in table 4.1 for German and 4.2 for English.

The results for the segmental model show that all three stylizations are able to improve the $F_0$ model with the local gradient stylization getting closest to the original $F_0$ contour. The simple extrema parameters, which do not contain information about the direction of $F_0$ movement, perform worst. Support points, which do have a directional component, perform similar to local gradient for German but worse for

| | training | | test | |
|---|---|---|---|---|
| $F_0$ stylization | segmental | combined | segmental | combined |
| baseline ANN | 0.1772 | N/A | 0.1818 | N/A |
| extrema | 0.1017 | 0.1373 | 0.1055 | 0.1495 |
| local gradient | 0.0787 | 0.1542 | 0.0817 | 0.1639 |
| support points | 0.0786 | 0.1732 | 0.0817 | 0.1817 |

Table 4.1: *Evaluation of $F_0$ features for the word prosody contour for German. MSE of normalized $F_0$ between natural prosody and prosody produced with computed word prosody contour, as well as between prosody produced with computed and expected contours.*

| | training | | test | |
|---|---|---|---|---|
| $F_0$ stylization | segmental | combined | segmental | combined |
| baseline ANN | 0.1733 | N/A | 0.1840 | N/A |
| extrema | 0.1007 | 0.1358 | 0.1079 | 0.1484 |
| local gradient | 0.0796 | 0.1534 | 0.0858 | 0.1634 |
| support points | 0.0901 | 0.1635 | 0.0962 | 0.1715 |

Table 4.2: *Evaluation of $F_0$ features for the word prosody contour for English. Normalized MSE between natural prosody and prosody produced by the segmental model as well as between prosody produced with computed and expected word prosody contour.*

English due to the higher ratio of mono-syllabic words for which the chosen syllable granularity for the support points is not sufficient.

Comparing the results for combined models, it is evident that for all stylizations there is still a relatively large gap between the expected word prosody contours and the ones learned by the tree-derived model. However, it can also be seen that a stylization is the more difficult to learn the closer it is to describing the original contour. The extrema can be learned best. The additional directional information is much more difficult to derive, possibly because it is less suited for the compositional approach of prosody creation. Still, some learning is possible so that the local gradient stylization still performs best overall.

### 4.4.2 Duration

The stylization of duration has been researched much less than that of $F_0$. There has been some research about theoretical models for speech rhythm in linguistics. An overview can be found in [RNM99]. However, this research cannot be directly transferred to a practical application on the speech signal and consequently has found little echo in speech synthesis, which is why it will not be discussed in more detail here.

In speech synthesis, duration is conventionally defined through the realized length of segments (phones and pauses). Proposed models differ primarily in the way these durations are normalized. Extended INTSINT [Hir99] uses categorical values to describe lengthening. Wightman proposed a mean/variance-based normalization that compensates for speech rate using a linear scale factor [WSHOP92]. Finally, van Santen introduced a model that computes intrinsic duration using a multiplicative model where each factor represents the influence of a prosodic control factor [vSKKM05]. He argues that any normalization based on simple mean durations results in a bias because the phonemes are not evenly distributed in the text and some may occur more often in contexts that are conventionally lengthened for prosodic reasons.

For the word prosody contour, the duration of the entire word needs to be described. The absolute length has obviously little meaning in this case as it is highly dependent on the phonetic content of the word. Even a normalization based on the average phone length in the word is bound to be correlated with the content because of the different intrinsic duration and the different elasticity of each phoneme. The stylization needs to describe the word duration more in terms of a phoneme-independent local speech rate.

Another effect that needs to be taken into account is that the speech rate within a word cannot be assumed to be constant. For example, linguistics have identified two important effects related to phrasing: pre-boundary lengthening (cf. [WSHOP92]) and domain-initial strengthening (cf. [KCFH03]), which describe effects of segment lengthening before and after phrase boundaries and are confined to the syllables directly adjacent to the phrase boundary.

To represent the speech rate, the notion of *expected duration* of a phone sequence is introduced. For each phoneme $p$, its speaker-specific expected length $\hat{l}_p$ is computed as the average length over all samples of the phoneme in the training corpus. Given a phone sequence $(p_1, \ldots, p_N)$ with $N$ phones, its expected duration $d_{p_1,p_N}$ is defined as

$$d_{p_1,p_N} = \sum_{i=1}^{N} \hat{l}_i$$

If further the realized length of each phone $p_i$ in the sequence is given as $l_i$ and thus the total realized length of a sequence is

$$L_{p_1,p_N} = \sum_{i=1}^{N} l_i,$$

then the *normalized lengthening* $D_{p_1,p_N}$ of the sequence shall be defined as

$$D_{p_1,p_N} = \begin{cases} 0.5 + \frac{d_{p_1,p_N}}{L_{p_1,p_N}} - 1 & \text{if } d_{p_1,p_N} < L_{p_1,p_N} \\ 0.5 - \frac{L_{p_1,p_N}}{d_{p_1,p_N}} + 1 & \text{else} \end{cases}$$

Following the ideas of the related work above, three different stylizations of word duration will be evaluated:

**Extrema** Similar to the $F_0$ extrema stylization, the parameters are meant to only describe the extrema between which the speech rate may vary. The speech rate is computed on a syllable base. The minimum and maximum of the normalized lengthening of all syllables of the word are used as word prosody contour parameters.

**Word average** This stylizations consists of only one parameter: the normalized lengthening of the entire word.

**ANN Durations** Like the word average stylization, it describes shortening/lengthening of the entire word. The difference is that expected phone lengths are computed using a segmental ANN model, which takes into account lengthening effects due to the segment context. The ANN is computed from the same training data prior to creating the prosody models and uses the same configuration as the duration ANN of the segmental model. The only difference is that word prosody contour features are omitted in the input feature vector.

To also take into account dynamics within the word, a variant of the latter two stylizations will be included where a gradient parameter has been added. The gradient is obtained using the normalized lengthening of each phone in the word. The lengthening is expressed as a function of the relative position of the phone, that is, for each phone $p_i$ we have a tuple $(\frac{i}{N}, D_{p_i,p_i})$. Then the gradient parameter is computed as the slope of the least-squares regression function over these data points.

The evaluation results for all five stylizations are listed in table 4.3 for German and 4.4 for English. The impact of the word prosody parameters on the overall concrete prosody is in general much smaller for duration than for $F_0$ because segmental properties have a much larger influence than functional prosody properties. Consequently, the difference between expected and computed word prosody contour is also much smaller.

The three different stylizations perform very similar. The most simple envelop parameters can be learned best but show a worse generalization behavior than the two averaging models. In terms of averaging, no improvement can be observed when predicting expected word lengths with the more complex ANN segmental model. The more simple mean durations are sufficient, possibly because the learning process is in itself an averaging process. The additional gradient parameter has a similar effect as for $F_0$ in that it is more difficult to learn from the tree-derived model. A much smaller but still relevant improvement can be achieved.

### 4.4.3 Pauses

Pauses present a special case because they are not directly a property of the prosody of the word. They are just an additional segment optionally added between words. They should nonetheless be counted as part of the prosody because they represent the strongest form of phrase boundary. For this reason and because the presence of a pause has a strong influence on the prosody of adjacent words, they are also included as a feature of the word prosody contour.

There are three factors that have an influence on pauses: the syntax structure of the sentence, rhythmic considerations like the length

| | training | | test | |
|---|---|---|---|---|
| duration stylization | segment | combined | segment | combined |
| baseline | 0.188 | N/A | 0.1915 | N/A |
| extrema | 0.1678 | 0.0778 | 0.1692 | 0.0838 |
| word average | 0.1697 | 0.0763 | 0.1715 | 0.0833 |
| word avg. + grad. | 0.1603 | 0.0940 | 0.1619 | 0.1023 |
| ANN durations | 0.1690 | 0.0764 | 0.1708 | 0.0835 |
| ANN dur. + grad. | 0.1608 | 0.0921 | 0.1623 | 0.0991 |

Table 4.3: *Evaluation of duration features for the word prosody contour for German. Normalized MSE between natural prosody and prosody produced by the segmental model as well as between prosody produced with computed and expected prosody contour.*

| | training | | test | |
|---|---|---|---|---|
| duration stylization | segment | combined | segment | combined |
| baseline | 0.1911 | N/A | 0.1969 | N/A |
| extrema | 0.1671 | 0.1006 | 0.1723 | 0.1086 |
| word average | 0.1688 | 0.1018 | 0.1733 | 0.1089 |
| word avg. + grad. | 0.1594 | 0.1154 | 0.1647 | 0.1211 |
| ANN durations | 0.1681 | 0.0970 | 0.1735 | 0.1081 |
| ANN dur. + grad. | 0.1595 | 0.1120 | 0.1649 | 0.1222 |

Table 4.4: *Evaluation of duration features for the word prosody contour for English. Normalized MSE between natural prosody and prosody produced by the segmental model as well as between prosody produced with combined and expected prosody contour.*

of the preceding phrase and finally segmental properties of the adjacent syllables, for example the presence of a glottal stop. The first two factors are mainly important to decide whether or not a pause is present while the segmental factors have an influence on the length of the realized pause. Accordingly, pauses are modeled similar to phone durations in two steps. Through the tree model the probability of the presence of a pause is computed and then a segmental ANN model is used to determine the length.

The word contour receives two additional features: pause before and pause after the word. The input features of the pause ANN model are composed of the word prosody contour of the preceding and following word and segmental features for the adjacent segments and syllables. The full set of segmental features is listed in appendix A.

### 4.4.4 Final Word Prosody Contour

Putting the evaluations of the previous sections together, the final word prosody contour is composed of 7-dimensional vector with the following content:

| | |
|---|---|
| $F_0$ | mean $F_0$<br>gradient<br>flatness |
| speech rate | stretching (relative to sum of average phone length)<br>gradient |
| pause | presence before word<br>presence after word |

## 4.5 Realization of the Mutator Function

The mutator function constitutes the basic element of the tree-derived model. For a given node in the syntax tree, it computes the contribution to the word prosody contour for the words in its subtree. The definition of the mutator function in (4.2) and (4.3) does not put any particular constraints on their design. Any statistical learner can be used. Special attention needs to be put into the choice of input features as it restricts the syntactic properties that can be taken into account in the model. The construction of the feature vectors will therefore be discussed in

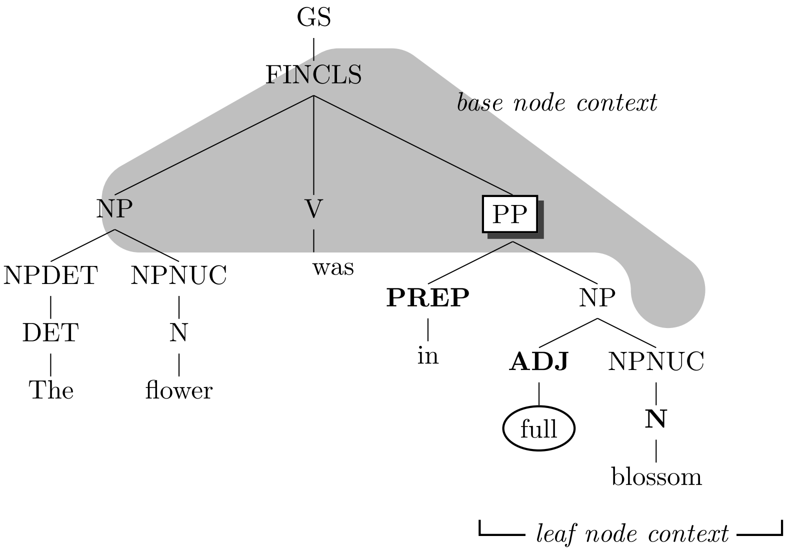


Figure 4.6: *Base node context and leaf node context taken into account for computing the word prosody contour for the word* full *by the mutator function of base node* `PP` *in the sentence* The flower was in full blossom.

detail at the beginning of this section, before the choice of statistical learners is explained and how they are applied in the mutator function.

### 4.5.1 Input Features

Statistical learners require a feature vector of fixed size. In section 4.2 we have already discussed that feature locality is assumed, thus limiting the context nodes taken into account. The feature vector $F_{\tau}(N, L)$ is defined as a composition of features from the *base node context* and the *leaf node context*. Figure 4.6 gives an overview over the nodes relevant for both contexts.

The base node context describes the local grammatical structure as realized in the syntax tree around the base node. The following nodes make up the base node context:

**Parent** defines the syntax structure in which the node is embedded.

**Child** is the node that is the direct ancestor of the leaf node. It defines the position within the node's subtree in a structural sense.

**Left/Right Neighbor** are the siblings that are directly adjacent to the base node. A node may not have a left or right neighbor when it is the first or last child within its parent. This is represented by a feature as well.

**Left/Right Siblings** are two unordered sets of nodes representing all siblings that precede and follow the base node, respectively. They express a more general context the node is embedded in.

The leaf node context describes the current position within the sentence on word level. It includes the following nodes:

**Leaf** describes the word the prosody contour is generated for.

**Left/Right Leaf Neighbor** are the words that are directly preceding or following. This may in particular be a punctuation mark which is an important indicator for adjacent phrase boundaries.

**Enclosing Leafs** are the leafs of the words preceding and following the subtree of the base node. Again the most important constituent types for these nodes are punctuation marks. Also important is the absence of such a node that indirectly encodes the beginning and end of the sentence.

For all these nodes their constituent and attributes are added to the mutator feature vector. Binary features are used to indicate presence or absence of a specific constituent type or attribute at a given position. In addition attribute features for the base node itself are added.

In addition to this structural information, the position of the base node within the tree has been added because phrasing has a strong rhythmic component (e.g. [**?**]), so that the presence or strength of a phrase boundary may depend on the length of preceding or following siblings. This is realized as the total number of siblings to the left and right of the node as well as their total length in number of syllables. Note that a relative instead of absolute positioning within the parent tree is used because interesting changes are more likely to happen at the beginning or end of the subtree. A similar observation was made by [BH05] who found more changes to happen at the beginning and end of prosodic units.

### 4.5.2 Construction of Feature Vectors

Feature locality reduces the number of nodes that need to be taken into account for the feature vector but the number of potential features is still large when all constituent types and attributes need to be encoded. The feature space can be further reduced, if the constraints of the underlying grammar are taken into account. For a given constituent in the base node, the constituent types that may appear in the node context make up only a small subset of all constituent types of the grammar. A similar observation can be made for attributes. Therefore, not one single statistical model is constructed for the mutator functions but a set of models: one model for each visible constituent type appearing in the grammar. When constructing the tree-derived model, for each node the statistical model is chosen that corresponds to the constituent of the base node.

Once the node context has been defined, the feature vector for each mutator function can be determined automatically. There are two possibilities: the grammar rules can be analyzed statically to compute which constituents appear at which position in the node context, or the syntax trees extracted from the training material may be analyzed. We have chosen the latter because it automatically eliminates structures that are not contained in the training data. For each mutator and each potential feature, its occurrences in the training sentences is counted and only those included where a minimum number of training examples can be found.

Note that this process does not eliminate highly correlated features. Correlation may in particular be found between attributes when they are passed on between nodes by means of the DCG attribute unification process. To alleviate the problem, a manual pre-selection of attributes has been made to exclude those that are only relevant for syntactic purposes. Still, the correlation between features remains high.

### 4.5.3 Random Forests for Mutator Functions

While in principle any regression learner is suitable for the mutator function, random forests [Bre01] were chosen in light of the difficult training conditions. Regression trees in general are well suited to cope with the kind of perfectly correlated features resulting from the automatic feature creation process as outlined above. Irrelevant features are further eliminated by the pruning process. Using random forests instead of single trees results in more stable results in face of the very sparse training data and speeds up the training process.

The mutator function needs to learn two different properties: first of all, whether a node is at all relevant for the prosody realization and second what kind of modification is done. These two properties can be unified if the output of the mutator function is not an absolute prosody contour but instead the modification to the contour is learned, thus

$$P_L^{k+1} = P_L^k + M_{N_k}(L, F, P_L^k)$$

Nodes that are not relevant for the produced prosody are represented with a mutation output of 0, a *zero mutation.* To that end, the output function of the standard classification and regression tree (CART, [BFOS84]) has been modified to yield a zero mutation state if the examples underlying each leaf do not sufficiently support the proposed prosody modification.

More formally, let $S = (s_1, \ldots, s_N)$ be the set of underlying samples that support a leaf. For one sample $s_i$, let $p_{s_i}^I$ be an element of the input word prosody contour and $p_{s_i}^O$ the expected output. Then the set of samples $E$ that support the trend set by the node can be defined as

$$E = \left\{ s \in S \middle| sgn(p_s^O - p_s^I) = sgn\left(\sum p_{s_i}^O - p_{s_i}^I\right)\right\} \tag{4.4}$$

The resulting prosody difference $\delta p$ of the leaf is then computed as

$$\delta p = \begin{cases} \frac{\sum p_{s_i}^O - p_{s_i}^I}{N} & \text{if } |E| > \theta N \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where $\theta$ defines the threshold for the percentage of samples that need to support the prosody modification. The higher a $\theta$ is chosen, the more pronounced a prosodic event needs to be, to be actually taken into account by the mutator. A value of $\theta = 0.8$ was manually set to be used for all experiments.

The random forest is made up of sets of modified CARTs. The number of trees per random forest varies according to the size of the input feature vector and the number of available training samples but is limited to a maximum of 200 trees. Each tree is constructed using 500 random training samples and the best split is determined over a set of 10 random features.

Zero mutations must also be taken into account in the final combination of the elementary trees in the random forest. Given the very sparse feature space we assume that a zero mutation result in a single CART is due to insufficient training material for the selected features and not evidence that the node is irrelevant for the final prosody output. A zero mutation of the random forest is thus only assumed if the number of CARTs that indicate a prosody change falls below a certain threshold. Otherwise the final output is computed as the mean over the output of all non-zero mutations.

To not further complicate the training process, each element of the vector that encodes the word prosody contour is computed independently. The final mutator consists of $w$ random forests, where $w$ is the size of the word prosody contour vector.

## 4.6 Training

The concatenation of mutator functions means that their training poses a decomposition problem. Only the final output of the word prosody contour is known. It needs to be decomposed into the individual contributions from each node before the individual mutator functions can be trained. The usual solution to this problem is a analysis-synthesis approach were intermediate outputs are computed by applying the mutator functions, the resulting global error is propagated back to correct the synthesized outputs and the new intermediate outputs are used to improve the mutator functions. Bailly uses this approach in his SFC model [BH05]. This error back-propagation proved to be problematic with the zero mutators used in the mutator model because during the back-propagation it is not known if a mutator has been correctly assigned a zero mutation, so that training errors can become self-enforcing. Instead a modified analysis-synthesis approach is used that is based on incremental tree building.

The target output of all mutator functions for the training is the expected word prosody contour of each leaf. Intermediate word prosody contours are only used for the input of the mutator function while the portion of the word prosody contour that stems from mutators applied later in the chain are considered noise. If no intermediate output is required, the tree of mutators can be built up gradually during training. The first round of training only takes into account either leaf nodes (for bottom-up application of mutators) or the root node (for top-down application). The training data is computed accordingly and the set of mutator functions with constituents available in this data are trained. Now the next level of mutators is added. The intermediate contours produced by the existing mutator functions provide the intermediate input for their input features. All mutator functions in this two-level tree are retrained. New levels are subsequently added until all syntax trees are used in full in the training. This approach guarantees that the input word prosody contour is always meaningful. The result is a more stable training and faster convergence.

To verify that the training against the final prosody contour provides sufficient results, a second version of the training is implemented where also intermediate values for the output contours are computed. This is realized by backward-computing the modifications made by subsequent mutators. Formally, given the sequence $M_0, \ldots, M_N$ of mutators applied for a leaf node $L$ where $\Delta P_L^k = M_k(F, P_L^{k-1})$ is the output of each mutator function, then the computed word prosody contour $P_L$ is defined as

$$P_L = P_L^0 + \sum_{i=0}^{N} \Delta P_L^i$$

Let further be $\hat{P}_L$ the expected word prosody contour of the leaf. Then the expected intermediate contour $\hat{P}_L^k$ for the training of the mutator function $M_k$ is calculated as

$$\hat{P}_L^k = \hat{P}_L - \sum_{i=k+1}^{N} \Delta P_L^i$$

This method shall be referred to as *forward-backward training.* Note that this is not the same as error back-propagation, as the error is still fully taken into consideration for each mutator.

### Handling Noisy Data

In later chapters we will discuss that the training data for the prosody models is prepared in a completely automatic way. This entails that the training material contains a number of errors. There may be parsing errors where the wrong syntax tree was produced. The phonetic transcript has mismatches with the signal where the speaker digresses from the script or the standard pronunciation. And finally, segmentation and $F_0$ estimation may produce errors in the annotation. Segmental errors are sufficiently locally confined to be considered noise. Word errors in the transcript and parse errors, however, have an effect on the correctness of the entire sentence. To compensate for these errors, a simple form of data filtering is applied for the training process: during each training round $t_i$, the global mean-square error $\overline{\mathrm{MSE}}_i$ over the entire training set is computed as well as the sentence-local $\mathrm{MSE}_i^k$ for each sentence $s_k$. In the subsequent training round $t_{i+1}$ all sentences $s_k$ with $\mathrm{MSE}_i^k > \alpha\,\overline{\mathrm{MSE}}_i$ are temporarily removed from the training set for that round. After the training , the new set of bad sentences is again computed over the full set of training sentences including those identified as bad in the previous round. The value for $\alpha$ was tuned manually and set to $\alpha = 4$ for all experiments in this thesis.

### Evaluation

For the evaluation of the different training methods the same setup was used as for the evaluation of the word prosody contours in section 4.4. As before, the evaluation at this stage is purely objective using the MSE of normalized duration and $F_0$. For a subjective evaluation, the reader is referred to chapter 7.

We compare top-down and bottom-up computation of word prosody contours as well as forward-only and forward-backward training. As before, the performance of the segmental model without the input from the word prosody contour is given as a reference. The results are shown in table 4.5 and 4.6 for German and English respectively.

| | $F_0$ | | duration | |
|---|---|---|---|---|
| Training method | training | test | training | test |
| baseline ANN | 0.1772 | 0.1818 | 0.1880 | 0.1915 |
| top-down, forward | 0.1633 | 0.1738 | 0.1819 | 0.1864 |
| top-down, fwd-bkwd | 0.1725 | 0.1766 | 0.1851 | 0.1886 |
| bottom-up, forward | 0.1617 | 0.1730 | 0.1808 | 0.1862 |
| bottom-up, fwd-bkwd | 0.1702 | 0.1756 | 0.1851 | 0.1891 |

Table 4.5: *Normalized MSE for $F_0$ and duration when using different training methods on the German corpus.*

| | $F_0$ | | duration | |
|---|---|---|---|---|
| Training method | training | test | training | test |
| baseline ANN | 0.1733 | 0.1840 | 0.1911 | 0.1969 |
| top-down, forward | 0.1653 | 0.1803 | 0.1847 | 0.1939 |
| top-down, fwd-bkwd | 0.1708 | 0.1818 | 0.1876 | 0.1937 |
| bottom-up, forward | 0.1629 | 0.1779 | 0.1840 | 0.1925 |
| bottom-up, fwd-bkwd | 0.1686 | 0.1811 | 0.1882 | 0.1946 |

Table 4.6: *Normalized MSE for $F_0$ and duration when using different training methods on the English corpus.*

For all tests, forward-only training shows better results than forward-backward training. It should be noted that the difference between the two methods is much smaller for the test set than for the training sentences, indicating that the forward-backward training generalizes better. However, forward-backward training also introduces an additional uncertainty during the backward computation which offsets this advantage. An additional error back-propagation might need to be investigated in future work.

Concerning the direction of contour computation, the bottom-up computation performs consistently better for the training data over both languages and both prosody parameters. For the test data, the same can be seen for English. The German corpus, however, no longer shows the difference. This may be due to the difference in the syntax structure of the two languages. German has a well-defined hierarchic clause structure while English is more structured as a sequences of phrases. Such sequences are more difficult to model in a meaningful way with a DCG grammar and the resulting syntax tree is slightly less informative with respect to prosody. However, when listening to the prosody produced by models created with the different training methods, it is noticeable in both languages that the top-down approach produces a much flatter prosody. Figures 4.7 and 4.8 show a typical example, here for the German sentence “In der Frühe weckt mich ein Chor von tausend Vogelstimmen.” Therefore, the bottom-up approach was preferred for both languages.

## 4.7 Discussion

In this chapter, we have introduced a novel method for prosody generation from the syntax structure of a sentence. The core idea is to decompose the prosody model into basic mutator functions that are combined according to the structure of the syntax tree. Such a model allows to represent arbitrarily complex sentence structures without preliminarily restrictions to the syntax properties used. The model is entirely data-driven. During the training process, the mutator functions not only learn how prosody is modified but also which syntax structures are relevant.

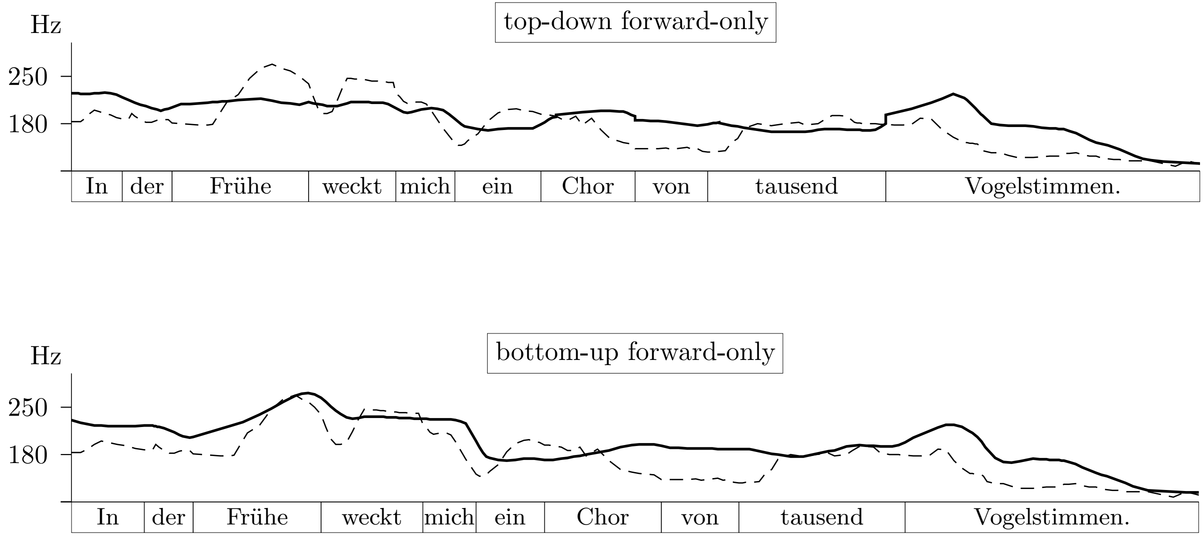


Figure 4.7: *Comparison of top-down vs. bottom-up application of mutators. The models have been trained with forward propagation only. The dashed line represents the natural $F_0$ contour as produced by a human speaker.*

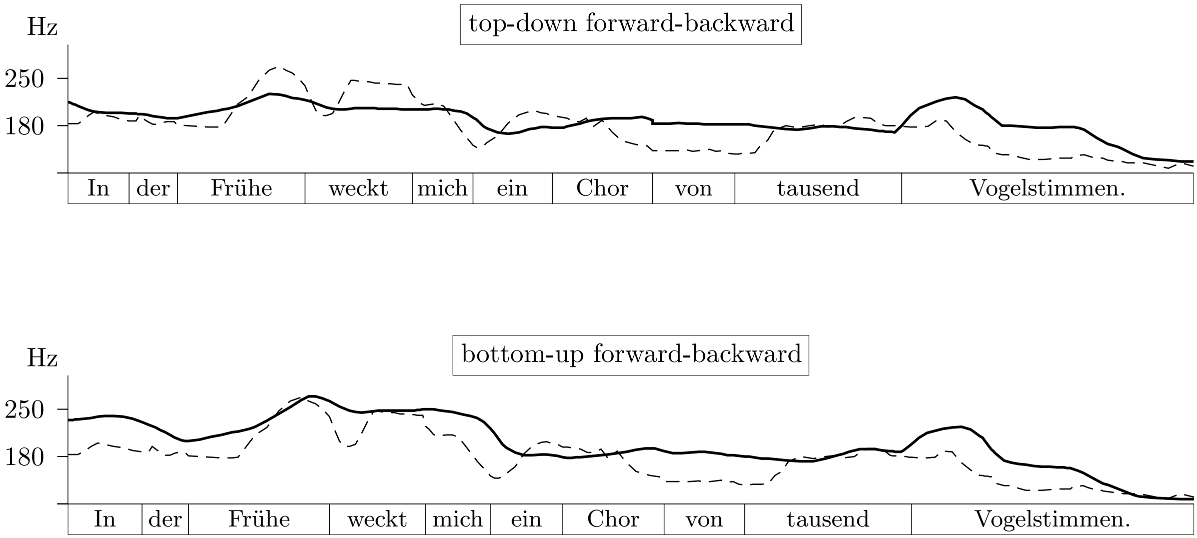


Figure 4.8: *Comparison of top-down vs. bottom-up application of mutators. The models have been trained with forward-backward propagation. The dashed line represents the natural $F_0$ contour.*

We have also shown that it is possible to learn directly the relationship between syntax and concrete prosody. Abstract prosody concepts like accents and phrases are learned by their realization rather than by defining them explicitly. Thus no prior knowledge about the prosody realization is necessary. This also implies that the prosody generation approach itself is language-independent. The only requirement for porting it to a new language is the existence of a grammar for that language and sufficient natural training material for the training of the model. The automatic preparation of the latter will be the topic of the next two chapters.

# Chapter 5

# Efficient Parsing of Large Corpora

Like every statistic approach to prosody modeling, the syntax-driven prosody generation needs to be trained on data of natural speech. With its high complexity and many degrees of freedom, a much larger corpus is required than for conventional models, in particular, if less frequent prosodic phenomena should be covered. As the goal is to generate prosody that expresses the discourse context to a certain extent, it is important that the speech has been recorded in a setting where the speaker has enough information to give the sentences an appropriate prosodic rendering. Consequently, prosodic speech corpora of isolated sentences are no longer sufficient, neither in terms of size nor in terms of context. Instead we aim to train the models on speech samples recorded in a natural setting outside the laboratory. There are many sources for such natural speech available nowadays, in particular from media resources. However, they can only be useful if they can be prepared and processed in an automatic fashion.

Chapter 5 and 6 take a closer look at automatic processing of large corpora not specifically recorded for speech processing. We will first discuss potential sources for training data and detail the audio book sources used as part of this thesis and for the evaluation in chapter 7. Then the complete process for corpus preprocessing is outlined. The remainder of this chapter focuses on text processing. The text normalization process as well as continuous parsing of long texts are discussed, which both together make it possible to quickly process arbitrary texts with little human intervention.

## 5.1 Training Material

Corpora that are specifically recorded for the training of voices and prosody have been dominant as training material for speech synthesis systems for many years. These corpora have the advantage that they have been recorded in a well-controlled environment by professional speakers and that they have been carefully reviewed to eliminate any mistakes. The result are highly consistent recordings with reliable annotations. However, for the syntax-driven modeling they are less suitable for two reasons: their small size and the lack of diversity.

The relatively small size of prosody corpora is a direct result of the cost- and time-intensive production process. Even though verification and annotation methods have improved in the recent years and can be largely automatized to reduce the amount of manual labor involved in post-processing, the preparation work and recording process itself are still labor-intensive. Consequently, such corpora can only cover a small range of prosodic phenomena and recording of large enough corpora in different styles is not feasible.

An even more important problem with prosody corpora is that they have been recorded with a neutral or artificial prosody to either reduce or enhance the influence of functional prosodic factors like emotions. This is even worse when recordings consist of readings of isolated sentences. Such sentences do not provide the reader with sufficient information about the context so that the resulting prosody is not sufficiently reflecting the potential relation between context and syntax as discussed in chapter 3. It is thus important that the training material for syntax-driven prosody modeling comes from continuously recorded speech.

Recordings of naturally produced speech have become available in large quantities in recent years: TV and radio shows, movies, audio books, speeches and presentations, just to name a few examples. They cover a wide variety of different communication situations and speaking styles and include many different speakers and languages, making them an ideal source for learning of a more diverse prosody. However, these resources also vary greatly in the quality of the recordings. Speakers may not articulate themselves very well, or they speak in a colloquial style or dialect which makes it difficult to obtain a transcript for these recordings. There may be background noise present and sections of overlapping speech during speaker turns. More important for the speech researcher is that the material is rarely annotated. In order to become suitable training material for prosody models, these corpora need to be preprocessed: relevant speech parts need to be isolated, text annotations need to be obtained and aligned with the speech, and prosodic information needs to to be extracted.

### Librivox Audio Books

With the focus of this thesis on reading of books, this and the following chapters will focus on audio books as suitable training material. In particular, we are using audio books from the Librivox project[1] for evaluation purposes. The project collects a growing number of nonprofessional readings of out-of-copyright texts and books in many different languages. All recordings are in the public domain.

The selected Librivox recordings have, like readings of books in general, a number of properties that simplify the task of preprocessing in contrast to the other speech sources listed above:

- Each of the selected audio books is read by a single speaker. There is no overlapping speech and no speaker detection is necessary. While the reading style may vary between books, a consistent prosody is used within the same book, so that no preselection of passages is required.

- The recordings consist of high-quality clear speech without background noise or overlapping speech.

- The text source used by the speaker is made available together with the recording, thus providing a highly reliable transcript of the reading.

[1] http://librivox.org

Regarding the last point, Braunschweiler [BGB10] analyzed an exemplary reading of Emma and found that the word error rate was 0.61 %, although this applies only for the reading of the book itself. Librivox recordings always contain a copyright preamble and an epilogue which is not part of the transcriptions. How these can be automatically removed will be discussed in section 6.2.

Even though the automatic annotation process is specifically optimized for recordings with these properties, it is in general also applicable to other types of media recordings, if additional steps for the selection of suitable recording sections are applied.

## 5.2 The Annotation Process

The training process as outlined in chapter 4 expects a training corpus that consists of sentences. More precisely, it expects for each sentence the following three resources:

**Syntax tree** the parse result with constituents and their attributes as well as the phonetic transcription for each word leaf,

$F_0$ **contour** a continuous curve of $F_0$ values[2],

**Segmentation** segmental durations where the segments must match the phonetic transcription of the syntax tree.

For the source material, only two resources are expected to be originally available: the audio book in form of one long continuous recording per chapter and the text of the book as a single continuous text file.

The annotation process needs first to partition the text and the audio recording into suitable sentence-like parts and then add the necessary annotations. In the following, such a completely automatic process for preprocessing of whole audio books is introduced. It is text-driven, meaning that first the transcription is processed and then the audio is aligned to the result. This allows to ensure that the partitioning of the book follows the parsing results and not the segmentation as chosen by

[2] Note that $F_0$ is also defined for unvoiced parts of the speech signal. In [Tra95] it was shown that the syllable-wise defined segmental ANNs yield better models, when $F_0$ is continuous instead of being defined piece-wise for voiced sections.

the speaker, simplifying the training process later. Figure 5.1 depicts the different steps of this annotation process.

As a first step, the transcript is normalized and parsed as a whole. The partitioning into sentences happens as part of this parsing process. It produces a sequence of syntax trees, which also contains the phonetic transcription of each word. Each syntax tree covers one sentence and thus the sequence determines how the corpus is divided into units. Next the audio recording is cut into the same sentence units as determined by the parsing by a applying an HMM forced alignment segmentation using special generalized phone models. After this step, any further processing can happen on these sentence units. That means that more precise algorithms can be used that are computationally not feasible on the long recordings of the original source. Those final annotation steps comprise HMM forced alignment yielding the segmental durations and the computation of the $F_0$ contour.

The text preprocessing stage with normalization and parsing will be discussed in more detail in the remainder of this chapter. Sentence alignment and phone segmentation are the topic of the next chapter. For the estimation of the $F_0$ contour, the $F_0$ annotation algorithm by Ewender et al. [EHP09] was used. It will not be further detailed here.

In the following, the terminology as established in figure 5.1 will be used. The unprocessed orthographic transcription will simply be referred to as the *text*. *Phonetic transcription* denotes the sequence of phones for a sentence as obtained from the leaves of the syntax tree. Finally, *segmentation* describes the same sequence of phones but augmented with positional information and position and length of pauses.

## 5.3 Text Processing

Syntax parsing is the first step of the annotation process. The parsing is based on the classic SVOX parser which is a multi-lingual direct-clause grammar (DCG) parser. This parser, however, was mainly developed for isolated sentences or short texts consisting of very few well-formed sentences. The analysis of literary texts holds additional challenges that need to be addressed.

First, the texts are much longer and syntactically very diverse. Short incomplete sentences may appear next to very long sentences

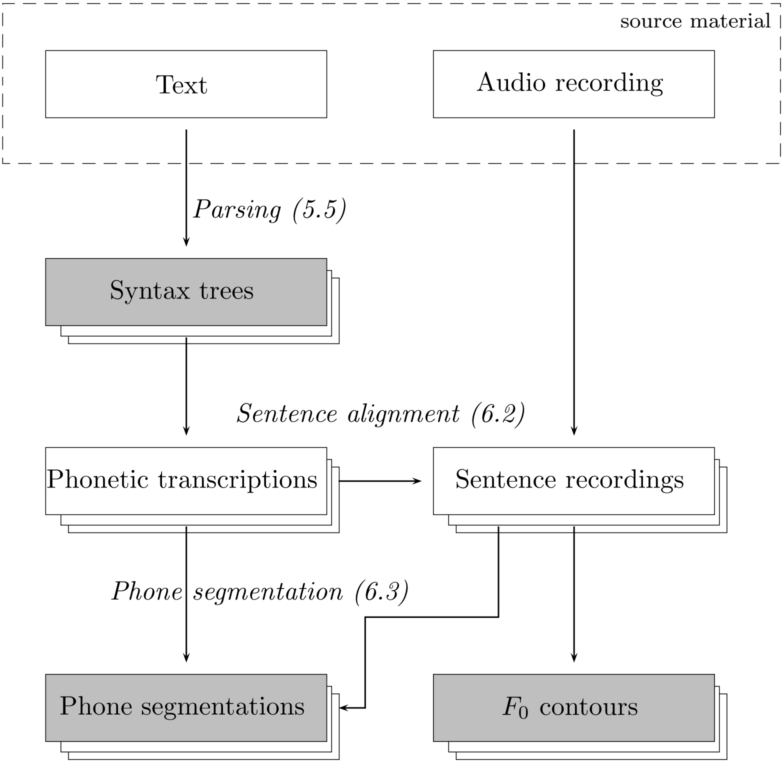


Figure 5.1: *Overview over the data flow during the preprocessing of natural speech resources. The resulting data consists of (in gray): syntax trees, segmentation and $F_0$ contours.*

that span entire paragraphs. Besides the simplifications and additions to the grammar as discussed in chapter 3, more efficient algorithms for the continuous parsing of such long texts are required.

Second, the formatting and the structure of these texts may contain important prosodic clues and should not be simply discarded. Structures like direct and indirect speech need to be parsed and the analysis should also be able to detect artistic expressions like ellipses and the creative use of punctuation. In addition, we cannot rely on the complete orthographic and syntactic correctness of the text. This is in particular an issue with the Librivox texts which originate mainly in the last century when grammar and spelling rules were applied much more liberally than in modern texts. While it would be possible to formulate at least part of these exceptions as simple grammar rules, a complete coverage of such formatting structures would increase the size of the grammar significantly and consequently reduce the parsing performance. Instead, an additional normalization module has been introduced that transforms the text into a standardized format before the text is forwarded to the parser.

The following section describes the normalization stage as used for Librivox books. Afterwards, the modifications to the parsing stage are discussed that are required to allow continuous parsing.

## 5.4 Text Normalization

While chapter 3 has discussed parsing of the syntactic structure, the text normalizer is responsible for processing the formatting, or *surface structure*, of the text. This encompasses the following elements:

**Text formatting** The format of the text defines how the text can be divided in chapters, sections and paragraphs. It furthermore describes how special structures like titles, quotes or lists are marked.

**Punctuation** In its normal grammatical function, punctuation is part of the grammar and parsed by the syntax parser. However, in fictional texts, punctuation is often also applied as a form of prosodic marking. How they are applied and what their actual meaning is, differs from writer to writer. As long as they are used in a consistent manner, these punctuation marks can be converted into prosodic markers to be applied in the prosody generation process.

**Quoted Speech** The most prominent special construct in literary texts is direct speech. Conventions how it is marked vary from language to language. Most common is the use of quotation marks and of special markers at the beginning of the paragraph. Processing of quoted speech is further complicated because an inquit may appear at any point in the quoted speech and because punctuation is often modified in the presence of quotation marks.

**Markup** Texts may also employ special markup for emphasized words or phrases and for inclusions. The two most common kinds of markup are special quoting and the change of font style like italic or all-caps style.

While most languages have an extensive set of rules for orthography and grammar, there are few fixed rules concerning the surface structure of the text. Consequently, each text employs its own set of rules, so that the text in itself may be coherent but texts are incompatible with each other when it comes to automatic parsing. The basic building blocks, however, are the same, thus the goal of normalization is to find and uniformly mark these building blocks.

The classic SVOX system follows a strip-and-parse strategy for the normalization of surface structure. Most markers of the surface structure like line breaks, markup and capitalization are removed. Punctuation beyond comma, full stop, exclamation and question mark is deleted as well. Orthographic ambiguities are resolved with an additional transduction step using a two-level rule automaton with hand-crafted rules ([Kos83], [Tra95]) to normalize differences in spelling. Finally, there are syntactic ambiguities like whether a full stop denotes the end of an abbreviation or a sentence. These are resolved directly during parsing [RP07]. This system is very robust against variations in formatting but at the price of discarding information that may be useful later for disambiguation of the text as well as the generation of prosody. Pushing much of the normalization into the transduction and parsing stage also has the disadvantage that it increases the complexity and, more importantly, the maintainability of the grammar.

To address these issues two additional concepts are introduced into the SVOX TTS: an additional normalization step and so-called *stage directions*.

Stage directions are commands that are injected into the text stream and passed through the processing pipeline where they can be further interpreted by the appropriate module. They can be used where the surface structure contains elements that cannot be processed by the syntax parser but are still necessary for the later prosodic processing stages. For example, style switches as they will be discussed in section 7.1 are implemented as a stage direction. Another possible use is for emphasis markers.

The normalization step transforms the text from its original form into a standard representation in terms of surface structure while trying to retain as much information as possible from the original formatting. It is further divided into the following sub-stages:

1. text structure (sections, paragraph, titles, quoting),
2. punctuation and character normalization,
3. style markers (quoted speech, emphasis, acronyms),
4. prosodic punctuation and capitalization.

Each step is realized independently and needs to be adapted to the text with a set of configuration parameters. The number of parameters is small enough that an automatic detection of the used surface structure is not worthwhile. In the following, the most important steps are described in more detail and the concrete modifications for the Librivox books used in the evaluation step are outlined.

### 5.4.1 Text Structure

The first step of normalization entails the detection of document structure. Paragraph, section and chapter breaks are detected and converted into stage directions.

In addition, special structural constructs need to be identified that interrupt the normal flow of text. For the audio books used in this thesis, two such constructs needed to be handled: section headings and poems. Other recordings may include additional constructs like lists and quotations. These special constructs are forwarded to the synthesis system as stage directions to change the text type. That means that if such a construct is found in the text a style switch stage direction is inserted and once it is finished, a stage direction is emitted to switch to the previous style. The prosody generation module can then either process them as style changes as they will be described later in section 7.1 or by interpreting them as additional input features for the mutator functions.

The concrete implementation of this step depends on the text format of the source. The transcripts for the Librivox recordings were available in text format. Section breaks were identified through section headings which in turn were detected using regular expressions that had to be adapted for each book separately. Poem sections could be identified through their particular use of spaces and line breaks.

### 5.4.2 Punctuation

Punctuation in the broader sense includes all non-alphanumeric characters that still appear in the text after the structuring stage. The function of punctuation marks can be grouped into three categories: a lexical function, for example to mark an abbreviation or an ordinal number, a syntactic function to delimit phrases and sentences and finally an emphatic function that serves as a hint for the reader how the text should be expressed. There is no one-by-one mapping between punctuation marks and function. Therefore, the actual function needs to be determined as part of the syntax parsing. A normalization is required because the same intended meaning may be realized with different punctuation marks by different authors. We therefore grouped the marks found in the different example texts and were able to identify four main groups as listed in table 5.1.

*Final punctuation* includes all sentence final markers. Apart from the syntactic placement, they may also appear in an emphatic function in the text. Either they may be repeated in that function ("How dare you?!") or they appear individually in the middle of the sentence ("Oh! dear, no, never." – Emma).

The three *non-final punctuation* marks have all a purely syntactic function albeit with a subtly different meaning. A comma always

| Final punctuation | Embedding |
|---|---|
| full stop | beginning mark |
| exclamation | end mark |
| question | boundary mark |

| Non-final punctuation | Emphatic punctuation |
|---|---|
| comma | exclamation |
| semi-full stop | ellipsis |
| segmental | pause |

Table 5.1: *Punctuation types processed by the syntax grammar. Normalization maps all punctuation to one of these types.*

denotes a phrase boundary in the regular grammatical sense. Semi-non final mark (usual a semicolon) is a stronger boundary and may be used in the place of final punctuation as much as in the place of a comma. Finally, the segmental mark (usually a colon) may only appear in sentence preposition or postposition but never inside a clause. Distinguishing between those three sub-types mainly helps to resolve ambiguities during parsing.

*Embedding punctuation* marks labels and proper names but may also function as an emphasis mark. If distinct markers are used for beginning and end, detection of embedded phrases is straightforward. However, many texts use the same mark for beginning and end. If this *boundary mark* is then represented by the same character as the apostrophe, it may even be confused with lexical punctuation. Disambiguation needs therefore to be done during parsing for these cases.

*Emphatic punctuation* is purely prosody related and has no relevance for syntax parsing. Consequently, it is implemented as stage directions and then deleted entirely from the input stream. This kind of punctuation that is used as a prosodic marker is most difficult to identify because the styling is entirely left to the preferences of the author. The text needs to be checked manually for regular occurrences and their format. The normalization stage can only define the categories that will later be used by the synthesis process.

### 5.4.3 Quoted Speech

The identification of quoted speech is the most important preprocessing step because there is a significant difference in the prosody style of quoted and narrative speech in book readings. At its simplest, quotation marks should trigger a style switch stage direction between normal and quoted speech. In practice, however, quotation marks are often used ambiguously. Doran [Dor96] lists many of the corner cases and concludes quoted speech can only be correctly identified by means of parsing. While parsing certainly would be able to achieve a higher correctness, including it in the normalization process has two other important advantages: first, as normalization works only with the surface structure of the text, the identification as part of the normalization step can be realized in a language-independent way. When included in the grammar, separate rules would have to be added for each new language. Second, the grammatical structure of sentences is the same for quoted and narrative speech. To make the parser aware of quoted speech, it would be necessary to duplicate many of the grammatical rules. If the parser remains unaware of style switches, the same rules can be used within and outside quoted speech sections.

How quoted speech is marked can differ significantly between texts. In particular, the following cases create ambiguity that need to be resolved on a surface level:

- If the text does not distinguish opening and closing quotes (for example, when using simple double quotes for both), the function has to be guessed. A simple heuristic based on the placement of spaces around the quote proved to be sufficient for the evaluated texts.

- Texts may use the same quotation marks for embedded phrases and quoted speech. However, while quoted speech can only be found at the beginning or end of a sentence, embedded phrases appear mostly in an unambiguous position in the middle of the sentence.

- Quotes may be nested in texts that tell a story inside the story or when a protagonist in the story quotes another person. In these cases only the outermost quotes have been taken into account

> because nested quoting is not foreseen by the prosody generation and would require further investigation to understand how it is realized.

Another particularity of quoted speech are inquits that announce the speaker in quoted speech. The boundary between inquit and quote cannot be simply modified into a style switch because the two parts may interact with each other in terms of syntax, so that the boundary would leave one part or the other in a partial state.

Inquits may appear before or after a quote or they may interrupt it. The three cases need to be handled differently.

If an inquit precedes or follows quoted speech, then the quote as a whole functions as an object to the verb of the inquit. The inquit itself may become very complex, as these two examples from *Emma* illustrate:

(1) But never did she go without Mr. Woodhouse's giving a gentle sigh, and saying, "Ah, poor Miss Taylor! She would be very glad to stay."

(2) "And really, I do not think the impression will soon be over," said Emma, as she crossed the low hedge, and tottering footstep which ended the narrow, slippery path through the cottage garden, and brought them into the lane again.

To ensure that the inquit sentence remains complete, the normalization process inserts a quote placeholder at the appropriate place in the inquit in addition to the style switch. For example

(3) "I want to go now!" she said.

will be converted into

(4) ⟨quote⟩ I want to go now! ⟨narrative⟩ # she said.

where terms in angle brackets mark a style change and # is the quote placeholder. Note that while "she said." may be parsed as a complete sentence, there is a difference in prosody between the complete sentence and the inquit sentence. Thus, the placeholder also serves as a marker for prosody generation.

If the inquit interrupts quoted speech, the two parts cannot be regarded separately because the quoted speech needs to be parsed as a whole, for example:

(5) “Emma knows I never flatter her,” said Mr. Knightley, “but I meant no reflection on any body. [...]”

The solution to this constellation is an inversion of quoting: the quoted speech being considered as the full sentence and the inquit as a special kind of embedding. That means that the normalization process needs to preserve quotes around mid-sentence inquits. There is only little change required in the grammar rules to take into account these kind of constructs as these inquit embeddings may appear at the same places as normal embeddings.

Finally, quoted speech involves a partial transposition of punctuation. Most notably, full stop is replaced by comma if an inquit follows. The normalization process detects and corrects the punctuation to comply with rules of unquoted speech.

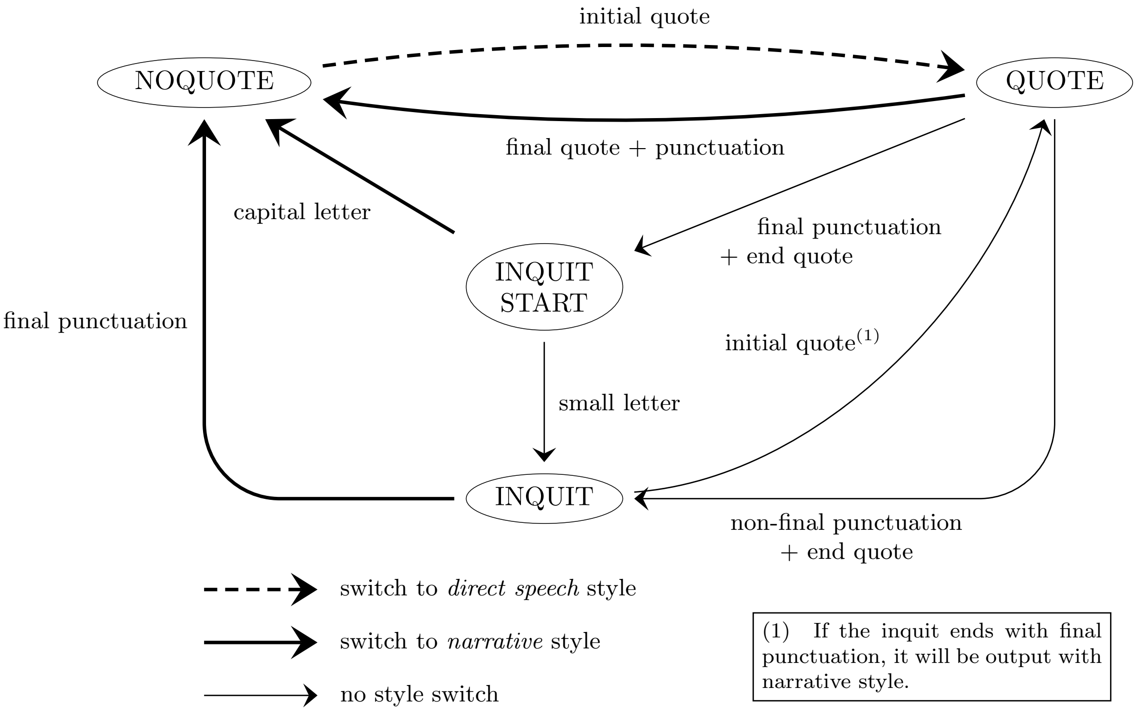
The full algorithm for determining direct speech is implemented as a state machine. A simplified version is shown in figure 5.2.

## 5.5 Robust Continuous Chart Parsing

After the text has been normalized, it can be parsed. The existing PolySVOX parser is a three stage DCG parser that already provides the mechanisms necessary for parsing large texts. However, the implementation has computational and memory constraints that do not allow to process longer sentences as often found in fictional texts. This section describes the changes made to the parsing strategy to ensure a memory-efficient real-time parsing of arbitrary texts.

### 5.5.1 The PolySVOX Parser

Before describing the modification to its implementation, this section gives a short overview of the original SVOX parsing process. Details can be found in [Tra95], [RP06] and [Rom09a].



(1) If the inquit ends with final punctuation, it will be output with narrative style.

Figure 5.2: *State machine used for the detection of quoted speech.*

Parsing is done in three steps: tokenization, word parsing and sentence parsing.

Tokenization splits the incoming text into lexemes as listed in the dictionary. To account for lexical variants, an additional normalization happens at this point. The text is transformed using a two-level rule (TWOL) automaton which transforms the surface structure of the text into an equivalent lexical structure that corresponds to the entries in the dictionary. Lexemes are extracted continuously. Further, the tokenizer checks for a *definitive word boundary.* Such a boundary is set when there is no lexeme that crosses the current character position. As soon as a definitive word boundary is found, all lexemes between this and the previous definitive word boundary are passed on to the word parser.

The word parser is implemented as a standard chart parser [Kay82]. Lexemes are entered into the chart by character position and then parsed bottom up. If a word cannot be parsed, the out-of-vocabulary (OOV) analysis is enabled. This analysis is implemented with DCG parsing as well. A special submorpheme dictionary and grammar exist for that purpose. They describe word stems on a submorpheme level. OOV analysis therefore means to repeat the word analysis with these special OOV rules enabled.

Words are parsed until a *definitive sentence boundary* is found. That is the case when all results of a word parsing step are sentence-final words or punctuation. At this point, all words are passed to the sentence parser which continues chart parsing using the sentence grammar. If no valid parse can be found at sentence level, an artificial syntax tree is created using a dynamic programming algorithm that finds the continuous sequence of constituents where the sum of penalties for all constituents in that sequence is minimal.

Multi-lingual parsing is made possible by loading multiple monolingual dictionaries and grammars at the same time and thus parsing the sentence in all languages in parallel. Small inclusion grammars also allow to define language switches mid-sentence, so that even mixed-lingual sentences can be parsed.

### 5.5.2 Efficient DCG Parsing

The main obstacle for parsing texts with long sentences is the explosion of the number of edges in the chart parser due to the multilingual

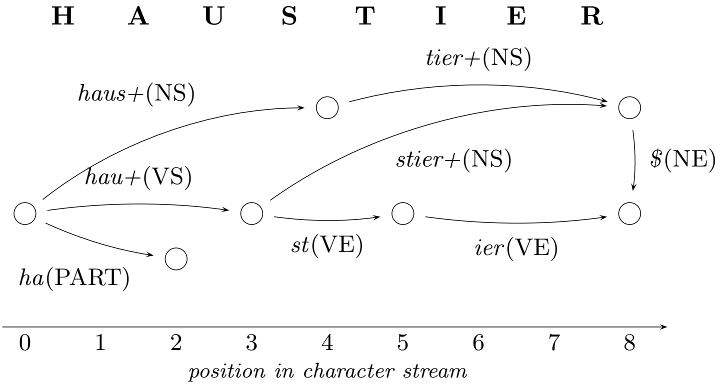


Figure 5.3: *Simplified example token network for the German word* Haustier*. The edges are lexemes, here given with their graphemic entry and constituent: noun stem (NS), verb stem (VS), noun ending (NE), verb ending (VE) and particle (PART).*

parsing. To be able to nonetheless constrain memory usage and computation time, the parser needed to be adapted to purge irrelevant parse edges as soon as possible. The following describes for each parse stage the modifications that were made.

### Tokenization

The first optimization is the reduction of tokens forwarded by the tokenizer to the word parser. The two-level rule (TWOL) translation from surface to lexical representation produces a network of lexemes where each node $n_i$ is uniquely represented by a tuple $(c_i, s_i)$ consisting of the character position $c_i$ in the input stream[3] and the state $s_i$ of the TWOL state machine. A simplified example of such a network is presented in figure 5.3.

Instead of forwarding all lexemes by their character position only, lexemes are forwarded by their network position $(c_i, s_i)$ and at the same time the network is pruned to remove lexemes that do not have

[3] In practise, a network node corresponds to a position between two characters in the input streams. Thus, character position $n$ denotes actually a node between characters $n - 1$ and $n$.

a direct successor. The input stream is processed continuously. At any point, the state of the tokenizer only consists of a set of currently open lexeme edges. These are edges which can be further extended to yield a complete lexeme. A single *parse step* appends a new character from the input stream and computes the new set of open lexical edges according to the following algorithm:

```
Data: set of open lexeme edges $L$, input character $c$
Result: new set $L'$, set of complete lexeme edges $R$
$L' \longleftarrow \emptyset, R \longleftarrow \emptyset$;
for $e \in L$ do
    if isLexemeComplete(e) then
        $S \longleftarrow$ setOfNewEdges(state($e$), $c$);
        if $S \neq \emptyset$ then
            $L' \longleftarrow L' \cup S$;
            $R \longleftarrow R \cup e$;
        end
    else
        $L' \longleftarrow L' \cup$ setOfExtendedEdges($e$, $c$);
    end
end
```

Note that completed lexeme edges are only forwarded, when a new open lexeme edge has been started at the end node.

The tokenizer further exports the notion of open nodes. A node is considered *open* if there is still an edge that might potentially start at this node. Otherwise it is *closed*. Thus, at the end of each parse step, the set of open nodes $O$ can be computed as $O = \{\text{startnode}(e) | e \in L\}$ where $L$ is again the set of open lexeme edges.

The resulting token network is an ordered graph, establishing a semi-order between the forwarded lexemes. All lexemes forwarded in a given parse step are guaranteed to appear after all lexemes that have already been forwarded in previous parse steps. This semi-order will be used by the word parser below.

### Word Parser

The notion of a lexeme network is kept within the word parser and the chart is built directly on top of this network, meaning that the chart is no longer linear but a directed graph as well. This has no implications on the parsing algorithm itself, which uses the same standard bottom-up parsing as PolySVOX. However, the notion of a definitive word boundary no longer exist because there may be multiple nodes per text character in the chart. Consequently, the word parser has been modified to continuously extend and parse the chart, as well as to continuously drop parts that are no longer needed.

The state of the word parser can be described by the triple $(N, E, \mathtt{W})$, where $N$ is the set of nodes as created by the tokenizer, $E$ the set of edges and $\mathtt{W}$ the constituent for edges describing complete words. Edges can be described by the triple $(n_s, n_e, c)$ where $n_s \in N$ is the start node, $n_e \in N$ the end node and $c$ the constituent of the edge. Further, for a given subset $X$ of edges, $N_X^s$ and $N_X^e$ shall describe the set of nodes that are start nodes and end nodes, respectively, of any edge in the set. Then each parse steps comprises the following:

1. Add and parse new lexeme set $R$.
   (a) Add new chart nodes: $N \longleftarrow N \cup N_R^e$
   (b) Add newly created lexeme edges: $E \longleftarrow E \cup R$.
   (c) Parse chart as far as possible.
2. Prune chart backwards.
   (a) Compute set $D = \{n \in N | \text{ closed}(n) \wedge n \notin N_E^s\}$ of prunable nodes, that is, nodes where no word can start in future parse steps.
   (b) Prune all nodes in $D$ and remove edges ending in those nodes.
   (c) Repeat (a) and (b) until no prunable nodes remain.
3. Extract newly parsed new words.
   (a) Compute $M \subset E$, the set of closed chart nodes where an already forwarded word has ended.
   (b) Find $W$, the set of word edges $(n_s, n_e, \mathtt{W})$ with $n_s \in M$ and $n_e \in \{n \in N | \text{ closed}(n)\}$.

   (c) Set $M \longleftarrow N_W^e$ to the set of end nodes of the newly found word edges.
   (d) Repeat (b) and (c) until $M = \emptyset$.
   (e) Forward all found word edges to the sentence parser.

4. Drop unused parts of the chart.
   (a) Determine set of unused nodes, see below.
   (b) Drop unused nodes and all edges ending in these nodes.

While step (2) causes the chart to be pruned backwards, (3) provides forward pruning by ensuring that only words are sent to the sentence parser that continue the already existing word network. Only approximately 7 % of the parsed words are then forwarded to the sentence parser.

Determining the part of the chart that can be dropped makes use of the property of semi-order of lexemes. For all networked nodes that are marked as closed by the tokenizer, the chart is also always closed. That means that these closed parts can only serve as a start node for edges that will be created in subsequent parse steps. Consequently, words created in later parsing steps cannot end at one of the closed nodes. However, they may still start at one of the nodes, if an existing edge constitutes the left-most child in a grammar rule. Thus, those partitions of the chart can be dropped where all nodes in the partition cannot be start nodes for a new lexeme anymore and they are not endpoints for edges that can still constitute the left-most child of a grammar rule that can still be applied.

To further compact the chart, packing of edges with the same signature [BL89] has been implemented. Packed edges are pruned as soon as the edge is guaranteed to be closed so that only the highest ranking solution is kept. Approximately 21 % of the edges can be removed immediately with this optimization.

### Backtracking for Unknown Words

Without an explicit word parsing step, it is no longer possible to repeat parsing of single words when no analysis is found for them. Instead, continuous word parsing requires a mechanism that allows to detect when unknown-word analysis needs to be enabled and the parser needs to be able to backtrack part of the process and redo parsing.

The word analysis may fail on two different points in the process:

- The **tokenizer** may not be able to produce lexemes from the text input. Formally, parsing fails when $L = \emptyset$, that is, there are no more open lexeme edges.
- The **word parser** fails when in step 3(a) $M$ is initially empty. That means that there are no more valid starting points for new words in the network.

If any of these conditions is detected, unknown-word analysis is enabled by switching on the OOV dictionaries and grammars.

To be able to backtrack, so-called *pick-up points* are introduced by the word parser. This is the set of nodes $P \in N$ where a complete word edge ended but no word yet begins, formally

$$P = N_W^e - N_W^s$$

where W is the set of forwarded word edges.

When the parser detects that the network can no longer be extended it restarts the tokenization process at one of the pickup points with OOV dictionary and grammar enabled. They are disable again as soon as the next valid word could be formed.

### Sentence parser

The sentence parser functions in a very similar fashion to the word parser. It differs in that sentences cannot be returned as soon as they are found because there is no paragraph grammar to determine the best fitting for multiple consecutive sentences. Instead, the parser here follows [RP06] more closely in that it searches for a definitive sentence boundary and determines the sentence or sentences at that point.

Parsing is done as soon as the word lexemes are passed from the word parser. For sentence end detection, again the fact is exploited that the words are in a semi-order. A forced sentence boundary is only assumed when all words returned in a parse step form a definitive sentence boundary. Whether a word edge is a definitive sentence boundary can easily be determined from the grammar. An edge that forms a sentence end can only appear at the end of rules that form sentence ends themselves. Thus all candidates for sentence ends can be determined recursively at the time the grammar is loaded.

Once a definitive sentence boundary is found, the sentences are determined. If necessary, artificial syntax trees are created.

### Constraints on Language Switching

Multi-lingual parsing puts a particular strain on computing resources because the language of a sentence is initially assumed to be unknown and therefore the bottom-up parsing needs to be done in all supported languages in parallel. The number of valid parses is even further increased by the inclusion grammars that allow switching between sentences. Many phrases that may be ungrammatical in one language may be parsable in another. Without further constraints, sentences may be created where all words are in one language but the word order is determined through the rules of another language. The PolySVOX parser tried to reduce these errors by introducing high penalties for language switching. However, this proved insufficient with literary text because of more complex sentence structures and the much larger dictionaries used. Therefore, the parser itself was made aware of the language of the lexemes in the dictionaries and of the grammar rules and the following two additional constraints were added:

- Given a direct path between the root of a syntax tree and a word leaf, the language can be switched only once on such a path. That means that a sentence may have foreign inclusions but the inclusions themselves must be mono-lingual.

- The parser may be given the base language of the text which will be preferred if results with equal weight are available in different languages. This mainly reduces the errors for short sentences and sentence ellipses.

In addition the number of inclusion rules was further reduced, so that switches are only possible on word level, within noun phrases or in form of embedded phrases.

## 5.6 Discussion

In this chapter, we have discussed text processing of entire books. The main issues are the large diversity of these text sources in terms of their surface structure and the large number of syntactic constructs that can be encountered.

For the standardization of the surface structure, we have proposed a number of normalization steps. This kind of normalization always involves a trade-off between preserving the information contained in the text structure and the ability to process the text automatically. The techniques used here rely for the most part on simple heuristics for the interpretation of the surface structure that are combined with a small amount of manual configuration. This approach allows to preserve more surface structure than the stricter normalize-by-discard mechanism used in PolySVOX but it also introduces new errors where the heuristics fail. For the preprocessing of the training material, such errors are easily tolerable as they only add noise to the training process. For synthesis itself, however, a more reliable mechanism will be required. This can either be done by moving some of the normalization steps to the parsing stage where then syntactic constraints can be used to resolve ambiguities or by manually preformatting the text, so that they are already in a canonical form.

The main challenge for parsing lies in the memory-effective handling of long sentences and paragraphs, in particular when combined with OOV analysis. With the modifications to the SVOX DCG parser, fast and continuous parsing of large texts is possible. All Librivox books could be parsed at an average of 62 words/s on a standard PC. This is well above the normal speech rate, so that the parser will also be suitable for real-time synthesis of large texts.

# Chapter 6

# Phone Alignment for Large Corpora

After discussing the processing of text in the last chapter, this chapter will focus on the processing of the speech signal, or more precisely, on phone segmentation. The phonetic segmentation of the speech signal does not just provide a simple annotation of phone durations. It is the most central part of the data preparation process because it provides the connection between the text and the audio recording. As such it is the means for linking any signal-related features to the text. Thus, the correctness and precision of the phone segmentation has a direct influence on the quality of all the prosodic models.

In the overview section 5.2, it was already discussed that the phone segmentation of large audio books is realized as a two-step process: the *sentence alignment*, where each sentence in the transcription is assigned its part of the speech signal, and *phone segmentation*, when the phone positions are computed precisely. This chapter will discuss both steps in detail. The next section gives a short introduction into the segmentation problem and provides an overview over the two-stage segmentation process. In the remainder of the chapter, we concentrate on two aspects of the segmentation: first a robust sentence alignment algorithm based on universal phone models and then a new method for improving the precision on conventional HMM-based phone segmentation.

## 6.1 Phone Segmentation from Text

The goal of phone segmentation is to mark the precise location of the phones in a speech recording. Given a speech signal and a sequence of phones that corresponds to the content of the signal, the most likely placement of boundaries needs to be found. In order to obtain a precise segmentation of a speech recording, an equally precise phonetic transcription of its content is required. Such transcriptions are generally not available for corpora not specifically recorded for speech processing. For the audio books covered here, only the original text is available, which is an orthographic transcript only and needs to be translated into a phonetic transcription first. Such translations are relatively simple to obtain. The text processing module of the TTS system can be used for this purpose or a statistical grapheme-to-phoneme (g2p) converter, as can be seen later in the evaluation of section 6.2. All these methods have in common that they only yield a canonical pronunciation of the text, not the actual pronunciation of the recording. Thus they introduce two kinds of potential errors: analysis errors and pronunciation errors. *Analysis errors* may occur for homographs when the parser chooses an entry from the dictionary with the wrong pronunciation as a result of a parsing error. They are even more likely to happen when unknown words appear in the text whose pronunciation has to be guessed. Romsdorfer [Rom09a] estimated a 15 % error rate for the pronunciation estimation of unknown words in polySVOX. *Pronunciation errors* appear when the speaker deviates from the standard pronunciation, either because there are allowable variants for a word or because of a slip. Both these error types are generally confined locally to a single phone or a short sequence of phones.

There is a third source of errors with text sources, *transcription errors*. They are the result of discrepancies between the transcript and what was read by the speaker. Those errors may become particularly severe when the transcript comes from a different source than the one used for reading. Different editions of the text may contain not only different wording but also have entire sections added or removed. Even when the original transcripts are available, discrepancies may appear when the speaker made mistakes while reading the text. These errors are mostly confined to word errors (single-word insertions, deletions or replacements) but may occasionally extend to missing or added sentences or even entire paragraphs. Thus, they have a more global effect on the segmentation result as the segmentation algorithm must be able to skip over longer non-matching sections. In case of the Librivox recordings used in this thesis, the transcripts are well behaved in the sense that they contain very few transcription errors that span more than a word.

In summary, an algorithm for phone segmentation of audio books needs to be able to tolerate these different error types while still achieving a high precision for the boundary placement of the parts of the transcription that are correct. These two goals are conflicting. To obtain precise boundaries, very accurate phoneme models are necessary that describe the exact extent of a phone. Error tolerance, however, requires that the models cover a wide variety of speaking variants.

As a secondary goal, the segmentation should be easily portable to other languages because no language-dependency exists for the syntax-driven prosody generation and should not be introduced through the data preparation step. This implies that it does not require additional external resources that are language-dependent.

### 6.1.1 Related Work

The general problem of the alignment between text and speech for large corpora has been researched for a wide range of material, from audio books to the annotation of TV or radio broadcasts. These resources differ in the quality of the recording and the transcript but they all share the common goal that a long continuous recording needs to be aligned to a text under the condition that there is no perfect agreement between recording and transcription.

The most common approach to this problem is based on automatic speech recognition (ASR). This idea was first introduced by [RRM97], subsequently made popular by [MJVTG98] and various improvements have been proposed in recent years ([MA09], [BGB10], [KBG$^+$11], [BCLM$^+$12]). The recording is first segmented into smaller parts and then piecewise processed with an existing ASR in order to obtain a hypothesized transcription. The language model of the ASR system is adapted to the available text in order to constrain the search space of the ASR. The hypothesized transcription and the text are then aligned with each other. The process of recognition and alignment is repeated with unsure sections until all text is aligned with a high probability of correctness. The alignment in the text domain is computationally much less expensive than an alignment between speech signal and text and allows large deviations between text and speech. Its main disadvantage is the dependency on a high-quality ASR system. Conventionally, a commercial ASR system is used, which have a sufficiently high recognition performance but which are only available for a limited number of languages.

Haubold et al presented a variation of ASR-based alignment that uses a simple phone recognizer without a language model to only detect well-recognizable phones [HK07]. They are then used as landmarks in the subsequent alignment process which still takes place in the text domain. Although their goal was mainly to align very imprecise transcriptions, it may also be suitable when no full-grown ASR system is available. However, they report a much higher error rate than solutions that use a commercial ASR system.

Another possible solution to the alignment problem for data in previously unseen languages are cross-lingual ASR systems, e.g. [LDY+09], [BSA+10]. Sufficient recognition results are reported only for systems that are adapted to a certain degree to the target language which in itself requires a significant training effort.

The output of ASR alignment methods is not a phone segmentation but only provides an alignment between the text and the recording. This still leaves open the problem of segmentation, that is, aligning the phonetic transcription of the text with the appropriate part of the signal. Thus, a second large group of segmentation algorithms aligns phone sequences directly with the recording. They are mainly based on forced alignment with HMMs [BFO93]. The advantage of HMM segmentation is that the models can be trained directly on the data to be segmented if a so-called *flat-start initialization* is used. Thus no external training material is necessary. Segmentation that uses flat-start initialization and adapted HMMs has been reported to be generally less accurate than when HMMs are used that have been trained on manually prepared data. [PKW96] reported better results by using a hierarchical method, segmenting into broader phonetic classes first.

The main issue with the HMM approach is that the computational requirements of the underlying Viterbi algorithm grow quadratically with the size of the speech recording and that segmentation works only efficiently if a precise phonetic transcription is available. In the presence of transcription errors, it has been reported to introduce gross misalignments, especially if pruning is applied, which is necessary for long speech signals. [STC$^+$03] attempted forced alignment using a hybrid approach where posterior probabilities obtained with a multi-layer perceptron are used as features for a HMM recognizer. In addition, they used a recognition network based on rules to model pronunciation variations. Prahallad [PB11] presented a solution that avoids pruning by segmenting the long speech recordings with a modified Viterbi algorithm that allows the speech signal to be longer than the text it is aligned against. Thus the text is segmented in small pieces one at a time.

Finally, some text-independent segmentation methods have been developed that rely on acoustic features only to determine potential phoneme boundaries. Examples of such methods are [AEM01], which proposes boundary detection based on jumps in the feature distance of acoustic features, and [GO07], which estimates boundaries by analyzing energy changes in different spectral bands. The advantage of text-independent methods is that they correlate much better with the dynamic changes in the speech signal. Boundaries are consistently hypothesized in dynamic parts of the speech signal resulting in very high recall rates. This comes at the price of over-segmentation. The problem remains to correctly assign the transcription.

### 6.1.2 Two-Stage Phone Segmentation

The robustness of ASR-based alignment stems mainly from its recursive refinement strategy where smaller and smaller sections are aligned, taking advantage of already established more coarse-grained alignments. In the following, we show that a similar strategy can be used with HMM forced alignment segmentation in order to overcome its problems of robustness and computational complexity.

The segmentation presented here is entirely based on HMM forced alignment but done in two stages as has been shown before in figure 5.1. In the first stage, the *sentence alignment* stage, the text is aligned with the recording on sentence level only. Sentence boundaries are normally accompanied by relatively long silences that can be unambiguously detected and do not require to be annotated with a very high precision. Thus, sentence alignment can be realized with a high tolerance against error. Once the sentences have been correctly identified, the recording can be cut at the annotated boundaries and the sentences then individually segmented further to phoneme level. This is the second stage, the actual *phone segmentation*.

Both stages are realized with HMM forced-alignment in order to avoid the use of language-specific ASR systems. The main difference between them are the models used. Sentence alignment makes use of generalized phone models that are highly robust against transcription errors, while phone segmentation relies on more precise models created from the corpus itself using Viterbi training with flat-start initialization. To further improve accuracy, we propose a signal-based boundary correction algorithm to be used together with this conventional HMM-based segmentation algorithm.

Note that this HMM alignment is guided by the transcription. The starting point are the phoneme sequences as received from the syntax parser. The sequences for the sentences are joined together resulting in a global transcription of the whole text. This global transcription is aligned against speech signal. Thus, the process presented here takes a somewhat reverse approach with respect to ASR-based alignment methods where the speech is first split on suitable silences and the text is then aligned against the speech segments.

The following section will discuss the sentence alignment and present evaluation results. Boundary correction for HMM segmentation will be discussed in section 6.3.

## 6.2 Sentence Alignment

The first step of the segmentation attempts to find a general alignment between text and recording. The SVOX parsing provides a partition of the text into sentence-like units in the form of a sequence of syntax trees. The goal of the sentence alignment is to split the audio recording accordingly and find the matching part of the speech signals for each syntax tree.

### 6.2.1 Alignment with Broadly-Trained Models

The most important property of the sentence alignment is robustness. The algorithm needs to produce correct alignments even in the face of larger transcription errors and, if misalignments happen, they must remain locally confined. This can be achieved by using forced alignment with broadly-trained HMMs, as will be explained in this section.

Forced alignment means that a fixed sequence of HMMs representing the phonetic transcription is time-aligned against the speech signal using the Viterbi algorithm. If the phonetic transcription corresponds exactly to the content of the speech signal, the overall alignment can be expected to be correct. How detailed the HMM describes a phoneme will only influence the accuracy of the boundary placement between phones. The more precise the HMMs, the better the accuracy of boundaries.

If the transcription deviates from the spoken utterance, as it is the case for the audio books, the alignment is no longer guaranteed to be correct. The erroneous parts of the transcriptions may align with wrong parts of the signal with a high probability. Due to the forced phone sequence, the overall error is normally confined locally if the search for the perfect alignment is exhaustive. This, however, is not possible for large speech corpora because the search space of the Viterbi algorithm increases quadratically with the length of the utterance to be aligned. The conventional optimization to be employed is pruning, meaning that only solutions within a certain distance of the best solution are kept (beam search). If the transcription deviates from the audio recording, the desired solution may have a very low likelihood locally, thus the right solution is more likely to be pruned prematurely. Very precise HMM sets, such as those used in phone segmentation, make the problem worse because the difference in likelihood between correctly and incorrectly aligned HMMs is the larger the narrower the HMM is trained. For a more error-resistant alignment a generalized set of HMMs is needed. If the models are more broadly trained, they are more likely to be aligned with partly diverging speech sections without being pruned too early.

Such broadly trained models come at the cost of accuracy and are therefore not usable for a precise phone segmentation. They are, however, precise enough to align text and speech on a sentence level. It was already said that finding sentence boundaries is a much simpler task because they are normally characterized by long silences. Furthermore, the required precision is much lower as it is sufficient to mark any point within the silence as the boundary.

### 6.2.2 Construction of Universal HMMs

The HMM set required for the alignment is different from HMMs used in speech recognition because we do not seek to obtain a precise description of a language. On the contrary, the goal is to construct a minimal HMM set that still allows text and speech to be aligned correctly. The HMMs only have a landmarking function that allows to identify the characteristic signature of the specific phone sequence of each sentence. Consequently, the HMM set does not need to be language-dependent. Instead we are going to construct a *universal HMM set* of phoneme-like elements that, once trained, can be used on arbitrary corpora in arbitrary languages without any further adaptation. For the design of this set, two aspects need to be taken into consideration: the choice of the phoneme set and the training material.

#### Phoneme Set

A model set based on grouping of phones has the advantage that mappings from language-specific phonetic transcriptions to phoneme groups are straightforward. The choice of phoneme groups has an important effect on the performance of the alignment. On the one side, the phonemes must not be too specific to avoid that the HMMs are trained too narrowly. On the other side, there must be enough differentiation for the phonemes to have the desired landmarking effect. In addition, the clustering should remain language-independent.

In order to estimate the influence of different phoneme classes, we tested phoneme clusterings on the audio book test corpora described below. A set of two classes, vowels and consonants, was considered the minimal configuration, then both classes were split until the alignment was sufficiently exact. Table 6.1 shows for selected cluster sizes the maximum deviation from the manually annotated boundaries, which is a good indicator of how well Viterbi is able to keep the overall alignment. The results indicate that a more fine-grained distinction is more

| Vowels/Cons. | 1/1 | 2/1 | 2/4 | 5/4 | 5/5 | 5/8 |
|---|---|---|---|---|---|---|
| ger | 30.5 | 30.3 | 30.3 | 1.3 | 1.3 | 1.3 |
| eng | 6.3 | 6.3 | 5.9 | 0.4 | 0.4 | 0.4 |
| fra | 3.5 | 3.4 | 0.8 | 0.9 | 0.9 | 0.3 |
| fin | 30.2 | 30.0 | 30.0 | 1.7 | 1.7 | 0.3 |
| bul | 27.9 | 23.0 | 23.9 | 16.4 | 16.4 | 1.4 |

Table 6.1: *Maximum deviation (in s) of sentence boundaries from their manually annotated position for different phoneme clusterings. The headings give the number of vowel and consonant groups in the phoneme set.*

important for vowels than for consonants, as increasing the number of vowel classes results in a much larger improvement of alignment than increasing the number of consonant classes. This is most likely because a consistent clustering across languages is much easier to find for vowels. Nonetheless, a more fine-grained clustering for consonants helped robustness in the cross-lingual case. For the final models the following set is used:

- vowels: open-front, open-back, neutral, closed, nasal, semi-vowel,
- consonants: plosive, aspirant, approximant, 4 fricatives,
- plosive pause,
- silence (sentence boundary).

Diphthongs are split and mapped to two vowel models according to the two parts they are made up of. Plosives are split as well into halt (plosive pause) and burst and mapped accordingly.

### HMM Training

The HMM set for the sentence alignment cannot be constructed from the training corpus itself at this point because flat-start initialization is not feasible for long recordings. It therefore needs to be trained from external speech material. To guarantee that the HMMs generalize sufficiently, they need to be trained on speech material that is sufficiently diverse, both in terms of speakers and language. For the experiments below, we used the recordings from the TIMIT database [GLF+93] to ensure speaker-independence and single speaker corpora in English, German and French (each approx. 1200 sentences by a female professional speaker) for a better generalization over different languages.

The signals are converted to 16 kHz and segmented using the automatic phone segmentation as will be described in section 6.3. The resulting HMMs use a feature set of the usual 12 MFCC features and their first derivatives together with energy estimated over a window of 24 ms. The phone models are linear HMMs with 3 states. Only the silence model is handled slightly differently to enforce a minimal length of the sentence boundary and avoid confusion with plosive pauses: it consists of 7 states that are all tied to one GMM. The HMM training and subsequent alignment were done using the HTK toolkit [You94].

### 6.2.3 Evaluation

The most important performance criterion of the alignment process is the correct placing of the sentence boundaries. A wrong placement means that a wrong transcription will be assigned to the sentence-like parts the recording will be split into after the alignment process. In the following, this performance will be evaluated under the aspect of robustness against transcription errors and under the premise of a language-independent application of the universal HMM set.

#### Test Corpora and Phonetic Transcriptions

In order to evaluate language-independence of the universal phoneme HMMs, an extended set of Librivox books was used that covers five different languages: German, English, French, Bulgarian and Finnish. The first three languages are already contained in the training material for the universal HMMs. Bulgarian and Finnish are used to test how well the universal phone set performs on languages unseen in the training data. The properties of the texts are summarized in table 6.2. The French and Bulgarian texts are read by male speakers, the remaining texts by female speakers. The recordings are divided in chapters of a length between 7 and 41 min, each chapter has been aligned separately.

| | | | | boundaries | |
|---|---|---|---|---|---|
| book | lang | m/f | parts | silent | non-sil. |
| hauff | ger | f | 6 | 1051 | 32 |
| emma-I | eng | f | 3 | 870 | 52 |
| Miserables | fra | m | 3 | 922 | 15 |
| Helsinkiin | fin | f | 3 | 776 | 21 |
| Legendi | bul | m | 3 | 851 | 9 |
| total | | | 18 | 4665 | 129 |

Table 6.2: *Librivox books used for the evaluation of the sentence alignment method: name of book, language, gender of speaker, number of chapters, number of sentence boundaries with and without silence.*

As we did not have dictionaries and grammars for all languages available, we used a different method to extract the phonetic transcription which goes as follows: for each language a grapheme-to-phoneme (g2p) model was trained using pronunciations extracted from the English version of Wiktionary[1]. These models were then used to convert the text transcripts into phonetic transcriptions using the statistical g2p conversion tool Sequitur G2P [BN08]. This method has the advantage that it is easily applicable to all five investigated languages. The disadvantage is that the resulting phonetic transcription has a much higher phone error rate (PER) than transcriptions obtained with a dictionary-based approach. When analyzing the quality of Wiktionary pronunciations in the context of speech recognition, [SOS12] found a PER between 3 % and 30 % depending on the language. Thus, the transcription used in the experiments below can be considered a lower bound on transcription quality. For the SVOX parser, the percentage of words not in the dictionary can give an estimate of the expected PER. For the books used in this thesis the proportion of unknown words is between 0.5 and 2.5 %,

In addition to the transcription, potential sentence boundaries had to be determined from the text. We used a naive method for this, placing sentences boundaries where there are final punctuations marks (period, exclamation mark, question mark) in the text. As punctuation is used rather liberally in the texts, this resulted in a number of

[1] `http://www.wiktionary.org`, a wiki-based open-content dictionary

assumed boundaries that were not actually realized as a pause by the speaker. These boundaries provide a much more difficult case for the alignment algorithm because the silence introduced at these boundaries become essentially a phone error in the transcription. In order to be placed correctly, the boundary needs to be placed exactly at the word boundary. Such *non-silent boundaries* also appear when sentences are determined from the syntax analysis, in particular when emphatic punctuation is used. For the books used in the evaluation in chapter 7 an estimated 5 % of sentence boundaries are non-silent. We therefore decided to not remove them and evaluate separately how the alignment algorithm performs with respect to them.

The alignment for all corpora was done with the same universal HMM set as described in the previous section. The Librivox books have the additional difficulty that each recording contains a preamble and epilogue section with title and copyright information for which no transcription is available. The exact wording varies between books and languages. While it would be feasible to remove these section manually, it is still a time-consuming manual processing step. Therefore, we added an additional skip model to the HMM set, which models speech in general. It was constructed as a 3-state HMM with one shared state whose parameters are computed over all available training material. This model was added at the beginning and end of each chapter, to allow skipping over preamble and epilogue. However, as these sections are relatively long, the pruning needs to be reduced in order to avoid misalignments. We found that a factor of ten is required to correctly skip preambles. Even with the reduced pruning, the chapters could be segmented in well under two minutes on standard PC hardware.

To be able to evaluate the alignment performance, the expected placement of the sentence boundaries was manually annotated for all test corpora. A boundary region was marked from the end of the final word of the preceding sentence to the beginning of the first word of the following sentence. Boundary placement by forced alignment was considered correct if the automatically determined boundary fell somewhere within this region. Non-silent boundaries were marked in the same way, so that the boundary region for them became very short, generally less than 10 ms. Next to the error rate, we also considered the deviation in case of an error which gives an idea how severe the additional error and consequently the misalignment between transcription

| | boundary errors | | deviation (in ms) | |
|---|---|---|---|---|
| language | silent | non-sil. | silent | non-sil. |
| ger | 4 (0.4 %) | 11 (34 %) | 1081 | 489 |
| eng | 1 (0.1 %) | 18 (35 %) | 354 | 193 |
| fra | 5 (0.5 %) | 7 (47 %) | 167 | 421 |
| fin | 1 (0.1 %) | 4 (19 %) | 261 | 76 |
| bul | 30 (3.5 %) | 4 (44 %) | 277 | 185 |
| total | 41 (0.9 %) | 44 (34 %) | 344 | 292 |

Table 6.3: *Boundary correctness of sentence alignment using unadapted models trained in three languages. Left are wrongly placed boundaries (absolute number and percentage within boundary class) and to the right is the average deviation of these wrong boundaries.*

and sentence recordings will be.

### Alignment Results

The results for the five test corpora are shown in table 6.3. For the intra-lingual case, the error rate for the silent boundaries is between 0.1 and 0.5 %. All boundaries are correct within 1.4 s of the manually annotated boundary. Manual inspection confirmed that all errors are confined to a deviation of one or two words. Thus, the number of additional transcription errors introduced by the sentence alignment is well below the WER of 0.61 % for the Librivox books as reported in [BGB10]. The error rate for non-silent boundaries is much higher, with an average of 37 % of the boundaries being shifted away from the true boundary. However, the deviation is equally low. This indicates that the errors are only caused by the imprecise nature of the HMMs which do perform worse in annotating precise word boundaries. The approximate position is still annotated correctly.

The two cross-lingual test corpora behave rather different. The Finnish text showed the best results of all test corpora. Finnish is a very regular language in terms of orthography. Therefore, the PER introduced by the statistical g2p converter can be expected to be much lower than for other languages. That in turn increases the performance of the HMM forced alignment. The performance penalty introduced by

| | boundary errors | | deviation (in ms) | |
|---|---|---|---|---|
| language | silent | non-sil. | silent | non-sil. |
| ger | 2 (0.2 %) | 10 (31 %) | 1102 | 389 |
| eng | 2 (0.2 %) | 15 (29 %) | 248 | 260 |
| fra | 3 (0.3 %) | 7 (47 %) | 340 | 353 |
| fin | 1 (0.1 %) | 4 (19 %) | 231 | 75 |
| bul | 3 (0.4 %) | 5 (56 %) | 892 | 345 |
| total | 11 (0.2 %) | 41 (32 %) | 602 | 250 |

Table 6.4: *Errors in sentence alignment and average deviation in case of error, when models are adapted for the corpus.*

the cross-lingual use of the HMMs is less important in this case.

The Bulgarian corpus performed significantly worse than the other corpora. One reason is that the corpora by male speakers did worse in general, possibly due to the fact that the training material for the universal HMMs was unbalanced with respect to gender. More importantly, Bulgarian is phonetically much less related to the training languages than Finnish. It features a rich set of palatalized consonants that are not present in either training language.

In order to compensate for the different phonetics, the HMMs can be further adapted to the specific speech recording. Using the phone segmentation obtained with the original alignment HMM set, a new model set is computed from the input corpus alone. Then the forced-alignment process is repeated with this new set.

The alignment results for the models so modified are shown in table 6.4. With this further adaptation step, the forced-alignment for Bulgarian performs as well as for the other languages. Then the overall error rate over all languages including non-silent boundaries is 1.1 %. However, the deviation in case of an error has slightly increased. The adaptation of the models not only moved the phone models closer to the language but also increased their precision. This immediately leads to a lower error-tolerance and consequently to a higher likelihood that the alignment process will fail. A possible strategy to counter-act this effect is to include some of the original training material when adapting

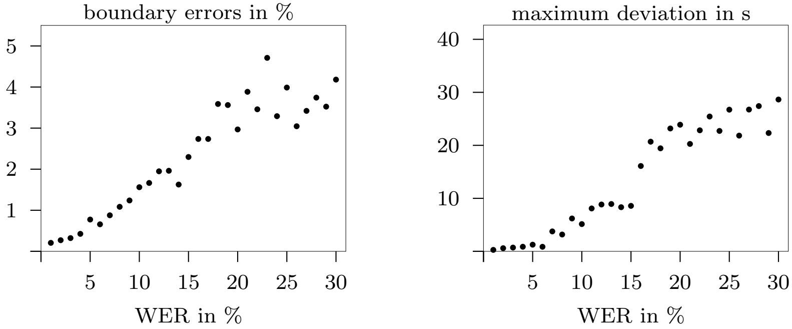


Figure 6.1: *Stability of sentence alignment with respect to word error rate. Percentage of misaligned sentence boundaries (left) and maximum deviation from true boundary (right).*

the models or using a different phoneme clustering better adapted to the language in question. However, this was not evaluated further as the performance was already sufficient for our application.

While the transcriptions for the Librivox recordings are generally very close to the audio recordings, such precise transcriptions may not always be available. To further test the robustness of the alignment when transcription errors are present, we artificially introduced word errors by adding and deleting words randomly. This corresponds to single word errors made by the speaker. The Finnish corpus was used for this experiment because it has performed best in the previous tests.

Figure 6.1 shows the number of errors and the maximum deviation for silent boundaries for increasing WERs. The algorithm is able to produce stable alignments up to approximately 5 % WER. At higher rates the deviation of misplaced boundaries increases significantly causing more than simple word errors when the corpus is segmented. At a WER of 15 %, the HMMs no longer can compensate for errors and alignment is partially lost.

## 6.3 Phone Segmentation

By splitting the recordings into sentences a corpus of short utterances is obtained which can be further aligned with the transcription using standard phone segmentation methods. In this second stage, the aim is to obtain phone boundaries that are as precise as possible. Like the previous stage, the segmentation needs to be able to cope with an imprecise transcription. However, a loss of alignment during the HMM forced-alignment will now be locally restricted to a single sentence so that robustness is less important and more precise models can now be used. With the shorter utterances, a self-contained alignment using flat-start initialized HMM phone sets is possible, so that this stage does not require any external training material for the models.

### 6.3.1 Motivation

Flat-start initialization means that initially phones are assumed to be evenly distributed over the signal and an initial phone set is computed using this segmentation. The phone set is then refined by iteratively segmenting the recording and recomputing the model parameters. Thus, the segmentation HMMs are trained as part of the segmentation process. As the process is unsupervised, it is less reliable than when HMM sets trained on manually segmented material are used.

An important reason for the worse performance of flat-started phone models in comparison to models trained on manual segmentation is that an unrestricted iterative embedded training is performed on data that is badly balanced in terms of phonetic context. The frequency of phoneme pairs in the corpus is determined by the language of the corpus. If a phoneme occurs very frequently in the same context then this fact is learned by the HMM by including part of the context in the model. During segmentation it results in a systematic error in the boundary placement. This effect can be visualized with a simple experiment. Table 6.5 shows the percentage of correct boundaries within a 5 ms and 20 ms deviation for a HMM-based segmentation using manually segmented data and the performance after the same models have been further trained for 40 iterations using embedded training. The boundary correctness has decreased significantly after the embedded training.

| HMM training method | Max. deviation 5 ms | 20 ms | Mis-aligned |
|---|---|---|---|
| isolated phoneme training | 51.13 | 89.50 | 0.59 |
| embedded training | 44.67 | 87.87 | 0.56 |
| flat-start init. + embedded tr. | 41.96 | 85.36 | 0.46 |

Table 6.5: *Segmentation performance for various HMM training methods on the TIMIT corpus. Percentage of correctly placed boundaries for different maximum deviations and percentage of misaligned labels, i.e. labels not overlapping with those in the reference segmentation.*

To avoid this issue, we enhance the standard flat-started HMM forced-alignment with an additional boundary correction step that corrects the boundaries using acoustic properties. Each boundary is moved to the place where the strongest acoustic break can be found in the speech signal. The phone models are flat-start initialized and iteratively trained using embedded training. A segmentation is produced via forced alignment and, at the same time, optional silences, plosive pauses and glottal stops inserted. The boundaries of the resulting segmentation are corrected, resulting in a preliminary segmentation that is the base for the next stage. The complete process is depicted in figure 6.2.

### Other Boundary Correction Methods

Various methods have been proposed to improve HMM-based segmentation. The largest group uses statistical models, for example, [ABGC05] proposes regression trees, [TGG03] applies statistical error models followed by a boundary correction using neural networks. Reporting between 92 % to 96 % accuracy within 20 ms, these approaches achieve close-to-human performance but they are not suitable for a fully automated segmentation approach, as the training of the statistical correction models again requires manually segmented training material.

A text-independent method for boundary correction that follows the idea outlined below more closely was presented by [KC02]. They propose to move boundaries to the place with the highest spectral continuity which is computed by comparing the spectral slope to the left

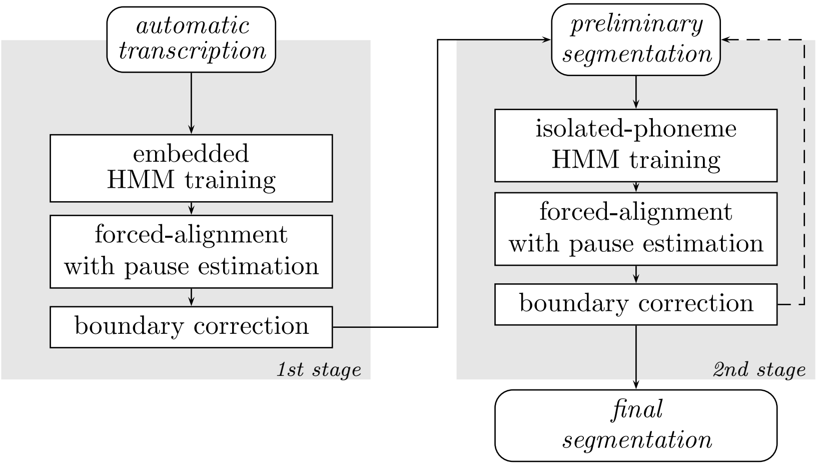


Figure 6.2: *Phone segmentation process. The first stage on the left prepares a segmentation, which in the second stage on the right is used for the iterative training of HMMs.*

and right of the potential boundary placement. Because this method produces a high number of spurious peaks, they restrict the correction to a time window whose extend depends on the phone class of the adjoining phones and is empirically determined. They report an improvement in accuracy from 93.1 % to 94.8 % (within 20 ms compared to hand labeling) for their test corpus.

In the next section, the details of the HMM segmentation process will be described before the boundary correction algorithm is introduced.

### 6.3.2 HMM-based Segmentation

The HMM segmentation still needs to be able to handle imprecise transcriptions. That means that the HMM parameters still need to be chosen in a way that the resulting models can tolerate minor errors. The most stable results are obtained with 5-state left-to-right linear models with features computed every 4 ms over a 20 ms window. This results in a minimum length of 20 ms for each phone.

Again, standard acoustic features were used: the first 12 MFCC coefficients, log energy and their first-order derivatives. Only one Gaussian mixture was used as we target highly consistent single-speaker corpora.

### Extending the Automatic Transcription

The automatic transcription determines the phoneme set present in the final segmentation and thus the set of phone models. Each phoneme is mapped to one phone model with the exception of diphthongs and affricates. Diphthongs contain movement between two phoneme regions, splitting them into two parts improves the boundary correction algorithm below. Affricates are handled like fricatives with plosive pauses in front.

Silences, glottal stops and plosive pauses require special handling. As the speaker has a high degree of freedom in how to realize them, they do not appear in the automatic transcription. They must be added to the segmentation for two reasons: first, they are required for the training of the prosodic models which should learn how they are realized. Second, they are very different from the rest of the phoneme set. Having separate models improves therefore both, HMM-based segmentation and the boundary correction.

The realization of silences and glottal stops can be learned and determined during the HMM-based segmentation. Silence models are initially trained from the silences at the beginning and end of the utterances, glottal stop models from mandatory glottal stops as transcribed. If no glottal stops have been annotated in the phonetic transcription, they need to be inserted based on heuristics. Here a simple heuristic was used placing glottal stops between each silence and vowel. This is sufficient for the English and German corpora used in this thesis but may have to be revisited for other languages. The actual realization of silences and glottal stops is then determined by introducing variants during the forced-alignment process. Optional silences are added after each word and optional glottal stops in front of each syllable that begins with a vowel.

### Plosive Splitting

The plosive-pause model cannot be trained like the silence model. If plosive pauses are added in front of each plosive for the initial training, the left context of plosives is restricted to plosive pauses only. This causes parts of the pause to be trained into the plosive models. Therefore, the plosive models are trained to contain the plosive pause in the first stage and phones are subsequently split into pause and plosion.

The splitting algorithm has to take into account that plosives may not be realized because of fusion or elision. In that case only a silence will be seen. It can also be observed, albeit less frequently, that the plosive pause is omitted, for example, after fricatives.

As the boundary between pause and plosion is normally visible over the entire spectrum, it is sufficient to search for jumps in the overall spectral energy. For each frame in the plosive, the difference in the spectral energy over a 30 ms window on both sides of the frame is computed. If the resulting function contains a peak and its maximum value is positive (meaning that there is silence before the frame and noise after it), the phoneme is split at the position of the maximum. Otherwise, a heuristic is used to determine the nature of the phone: if its mean energy is below the mean energy of the silences in the utterance, a plosive pause is assumed, otherwise a plosion. Finally, where the splitting process created any consecutive plosive pauses, they are fused together.

## 6.3.3 Boundary Correction

Phone boundaries can be considered as a transition phase between relatively stable centers of the phones. As such, they are very well visible in a feature-distance matrix. Figure 6.3 (a) shows such a matrix for the phrase *she took one.* This property can be used to locally correct the boundary between two phones.

Given two adjacent phones $p_i$ and $p_{i+1}$, the boundary is located between the perceptual centers of these phones. These centers shall be called *core frames*. Given the core frames $c_i$ and $c_{i+1}$, the ideal boundary is placed such that all frames to the left are acoustically closer to the left core frame and all frames to right are closer to the right core frame. Such a boundary does not necessarily exist, so we compute the

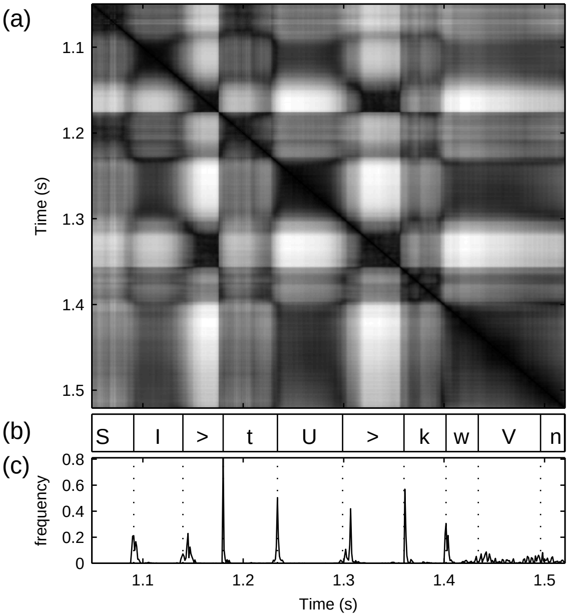


Figure 6.3: *Feature-distance matrix (a), reference segmentation (b) and distribution of boundary placement (c) for phrase 'she took one'.*

best boundary as follows: if $x_{c_i}, ..., x_{c_{i+1}}$ is the series of feature vectors between the two core frames, the two confident boundaries $\hat{b}_{i,i+1}$ and $\hat{b}_{i+1,i}$ from the left core frame and the right core frame, respectively, are computed as

$$\begin{aligned} \hat{b}_{i,i+1} = b \quad | \quad & D(x_{c_i}, x_b) \geq D(x_{c_{i+1}}, x_b) \wedge && (6.1) \\ & \forall\big[c_i < f < b : D(x_{c_i}, x_f) < D(x_{c_{i+1}}, x_f)\big] \\ \hat{b}_{i+1,i} = b \quad | \quad & D(x_{c_i}, x_b) \leq D(x_{c_{i+1}}, x_b) \wedge && (6.2) \\ & \forall\big[b < f < c_{i+1} : D(x_{c_i}, x_f) > D(x_{c_{i+1}}, x_f)\big] \end{aligned}$$

where D(x,y) is the Euclidean distance between two feature vectors $x$ and $y$. Then, the final boundary between the phones $p_i$ and $p_{i+1}$ is

$$b_{i,i+1} = \left\lfloor \frac{\hat{b}_{i,i+1} + \hat{b}_{i+1,i}}{2} \right\rfloor \tag{6.3}$$

If the core frames are located in the true center of each phone, the boundaries will be placed as expected: on jumps in the distance function, if there is one, and equidistant from both core frames otherwise. However, the position of the core frames needs to be estimated from the placement of the phones in the imperfect preliminary segmentation. In order to evaluate how strongly the boundary function depends on the correct choice of the core frames, we computed the boundaries for any two frames from two adjacent phones from the reference segmentation. Figure 6.3(c) shows the result for the example phrase *she took one.* There is a clear preference for frames close to the reference boundaries. Therefore, choosing any frame that is close to the center of the phone is sufficient as core frame.

Assuming that the HMM-based segmentation has placed the phones in the right vicinity, we compute the frame that is most typical of the ones in the phone, i.e. that is closest to all other frames. The core frame $c_i$ of a phone $p_i$ with the preliminary boundaries $b'_i$ and $b'_{i+1}$ is then

$$c_i = \underset{b'_i < f < b'_{i+1}}{\operatorname{argmin}} \Big( \underset{b'_i < f' < b'_{i+1}}{\operatorname{median}} D(x_{f'}, x_f) \Big) \tag{6.4}$$

| | | Max. deviation | | | Mis- |
|---|---|---|---|---|---|
| stage | step | 5 ms | 10 ms | 20 ms | aligned |
| 1st | after forced alignment | 40.23 | 66.61 | 86.61 | 0.93 |
| 2nd | after forced alignment | 46.39 | 71.58 | 89.06 | 0.66 |
| 2nd | after bnd. correction | 53.03 | 72.04 | 89.14 | 0.43 |
| 2nd | after 10 iterations | 53.47 | 72.70 | 89.45 | 0.29 |

Table 6.6: *Segmentation of German prosody corpus. Percentage of correct boundaries for different maximum deviations after HMM forced alignment in 1st stage and first iteration of 2nd stage and after boundary correction in the 1st iteration and 10th iteration of 2nd stage.*

where $D(x, y)$ is again the Euclidean distance. The median is used instead of the mean to be more robust against noise in the signal.

In contrast to the HMM-based segmentation, where relatively coarse features are required to minimize segmentation errors, the boundary correction yields the best results with short-term features computed every 1ms over a 10ms window. Experiments show that perceptional features give slightly better results. The first 12 PLP coefficients as computed by the HTK toolkit together with normalized log energy are used.

### 6.3.4 Evaluation

The segmentation process was evaluated against a German prosody corpus recorded in our group and against TIMIT [GLF$^+$93]. We evaluated the precision of the boundary placement on the one side, and the robustness of the segmentation on the other side, the latter by evaluating the number of misaligned labels, that is, segments that have no overlap in time with the corresponding labels from the reference segmentation.

#### Prosody Corpus Segmentation

The German prosody corpus consists of 186 sentences spoken by one male professional speaker sampled at 16kHz. A manual segmentation of the entire corpus was available from a previous project. The automatic transcription of the corpus is very close to the manual transcription.

Differences are mostly related to optional glottal stops.

Table 6.6 shows the segmentation performance for this German corpus. The corpus is very small for the training of the HMM. The phone models from the first stage generalize badly, which causes 0.9 % of the phones to be misaligned. The boundary correction allows better models to be trained in the second stage. This is why the iteration of the second stage does not suffer from diverging boundaries like embedded HMM training without correction does. The iteration process further reduces the number of misalignments by 47 % after 10 iterations.

The boundary correctness of 89 % within a 20 ms deviation is close to the performance of similar HMMs trained with manually segmented data reported by e.g. [BFO93] or [ABGC05]. Segmentation of a larger corpus is expected to perform equally well. With a correctness of 53 % for a 5ms maximum deviation the boundary correction clearly outperforms any HMM-based segmentation in terms of precision.

### TIMIT corpus segmentation

In order to evaluate the robustness of the segmentation with respect to imprecise transcription and be able to compare our approach with other segmentation algorithms, we also segmented the TIMIT speech corpus of American English. The corpus differs from the targeted audio book corpora in that it was recorded from multiple speakers without professional training speaking multiple dialects. We used all 1344 sentences from the test set spoken by 168 different speakers of 6 different dialect regions. To compensate for the resulting greater phoneme variability the phone models have been trained with 4 Gaussian mixtures. Otherwise the HMM configuration was the same as described in section 6.3.2.

We did two sets of experiments, the first using a manual, the second an automatic transcription. We used the standard TIMIT phoneme set as defined in the lexicon. Where the reference segmentation distinguished additional phoneme variants, they were mapped to their closest counterpart from the standard set.

The manual transcription was taken directly from the reference segmentation including silences, glottal stops and plosive pauses. The plosive splitting algorithm was applied nonetheless. The automatic transcription was produced with the TIMIT dictionary and the word list

| | | Max. deviation | | | Mis- |
|---|---|---|---|---|---|
| stage | step | 5ms | 10ms | 20ms | aligned |
| Manual transcription | | | | | |
| 1st | after forced alignment | 41.96 | 67.57 | 85.36 | 0.46 |
| 2nd | after forced alignment | 49.43 | 73.44 | 88.22 | 0.48 |
| 2nd | after bnd. correction | 54.26 | 77.09 | 90.23 | 0.40 |
| 2nd | after 5 iterations | 54.21 | 76.92 | 90.07 | 0.41 |
| Automatic transcription | | | | | |
| 1st | after forced alignment | 36.69 | 61.26 | 80.17 | 0.82 |
| 2nd | after forced alignment | 47.82 | 71.08 | 85.80 | 1.17 |
| 2nd | after bnd. correction | 53.48 | 75.58 | 88.40 | 1.16 |
| 2nd | after 5 iterations | 53.36 | 75.41 | 88.36 | 1.16 |

Table 6.7: *Segmentation of TIMIT corpus. Percentage of correct boundaries and percentage of misaligned labels after the different segmentation stages.*

provided for each sentence. If there was more than one transcription for a word in the dictionary, one was chosen randomly.

Table 6.7 shows the results for both experiments. Boundary correction improves the segmentation by 2.3 % relative for manual and 3.0 % relative for automatic transcription for a 20 ms maximum deviation. [KC02] reported only 1.8 % relative improvement, but on a segmentation with a higher baseline. The results are similar to those achieved on the prosody corpus. This shows that the correction works equally well for corpora with a higher variation in segmental quality.

For both experiments, the number of misaligned labels is reduced after boundary correction, making the method very robust against transcription errors. Additional errors are introduced by the HMM-based segmentation in the second stage, though, most probably, because the more precisely trained models are less flexible in skipping over transcription deviations. Increasing the Gaussian mixtures to allow more variation counters the effect in the case of manual transcription but not for the automatic one. Introduction of variants may be necessary instead.

The boundary correction performs better than the statistical correction proposed in [TGG03]. However, the boundary refinement based on neural networks in [TGG03] as well as the regression tree refinement investigated in [ABGC05] outperform our method in terms of total precision. This is not very surprising because statistical boundary corrections learn the particularities of the manual segmentation they are trained on while a text-independent correction method relies on properties of the signal only, with which a human labeler might disagree. Whether a higher agreement with the reference is desirable, eventually depends on what the segmentation is used for.

## 6.4 Discussion

The two segmentation methods discussed in this chapter complete the parts necessary for an entirely automatic preparation of training data for new prosody models. The sentence alignment provides a robust method to split the audio recordings into sentences according to the syntax tree output produced by the parser. It is able to reliably predict the boundaries between sentences even in the face of transcription errors and missing silences between sentences. For English and German an error rate of 0.2 % can be expected for sentence boundaries with a realized silence and 30 % in absence of a silence. Given an estimated 5 % of non-silent boundaries in the Librivox corpora when parsed with SVOX, an additional 1.6 % of word errors can be expected from the segmentation process.

A particular advantage for the Librivox books is that generalized models also allow to skip sections of speech using general speech HMMs, so that the non-transcribed preamble and epilogue do not need to be removed manually. The only manual preparation step that remains is the partitioning of the transcript into chapters to match the partitioning of the recordings as provided by Librivox.

Introducing an additional signal-driven correction step to the standard HMM forced-alignment during phone segmentation allows to achieve the same precision with a flat-started HMM phone set as with one that was trained on manually segmented material while remaining completely self-contained. Transcription errors are locally confined, so that the general alignment between phones and signal remains correct. This is especially important as the $F_0$ alignment with the text depends on the correctness of the segmentation.

As both segmentation steps use HMM forced alignment, there is no dependency on an ASR system. The generalized models of the alignment step still need to be trained on external training material. However, as the training data does not necessarily need to be in the same language as the recording to be segmented, the segmentation and thus the complete preparation can be considered language-independent.

What has not been addressed yet is the detection of pronunciation variants. If deviations from the standard transcription could be detected, an even higher precision in the placement of phone boundaries could be achieved which would further improve the prosody models for duration. Such variants could even be used during synthesis if the prosody model can also learn patterns in the use of the different variants.

# Chapter 7

# Reading Fiction

Putting it all together, this final chapter will address the application of the prosody generation in a TTS system for reading of fictional texts. The automatic data preparation as discussed in the previous two chapters is already an important prerequisite because it allows to easily make training material available that comes from a natural context and is sufficiently large to cover a wide variety of syntactic constructs. Yet the syntax may not always be sufficient for a lively reading of complete books. For a more realistic rendering, factors should be taken into account that go beyond syntax. The first part of this chapter will discuss three of these aspects: the application of different reading styles, the inclusion to additional hints of discourse structure and finally the extension to multi-lingual models. The chapter will then conclude by presenting the results of a perceptual evaluation of the generated prosody.

## 7.1 Speaking Styles

There is no single correct realization of concrete prosody. Every speaker has a certain freedom to adapt their speaking style to the situational context. For example, one might speak faster in a informal conversation than when giving a speech. However, in general the speaking style cannot be reduced to such simple parameters as speech rate. The concrete prosody changes as a whole. That is why it is important to take the speaking style into account when choosing the training material for the prosody model.

There are different factors that can have an influence on the choice of the prosodic style. The most important ones are

**Speaker habits** Different speakers realize the same speech very differently due to physical restrictions, personal habits and social influences from their environment.

**Emotional state** Emotions like anger, happiness or sadness cause the same speaker to change their speaking style in a way that allows the listener to identify these emotions. Research in emotional speech synthesis has however shown that listener agreement on detecting emotion is lower unless the emotions are overstated (cf. [GKMN07]). That indicates that the emotional style is strongly coupled with speaker habits.

**Conversational context** The speaking style is also adapted to the type of speech produced. Read speech differs from spontaneous speech, a political speech will require a different way of expression than a technical talk or a radio announcement.

Some of these factors, in particular emotional state, can in fact be considered as regressive factors. This makes learning from natural speech problematic because the training data needs to be annotated accordingly. Categorical factors like speaker and conversational context are better suited for a data-driven prosody learning process as they can be automatically determined. They are also much easier to assign when synthesizing speech.

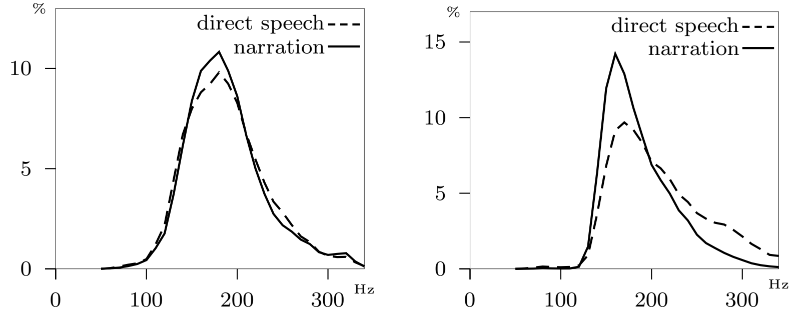
There are two basic approaches how the style can be taken into account in syntax-driven prosody generation. Either one or more style properties can be incorporated into the prosody model as additional features for the statistical models, or a completely different prosody model can be trained. Yamagishi compared the two approaches for synthesis of emotional speech with HMM synthesis and came to the conclusion that there is no significant difference in performance of the two approaches [YOMK03]. However, as a style influences the prosody on a global scope and as a style normally does not change with a sentence, the training of different models is the more straightforward solution in this case. In the following, the switch of conversational style will be discussed in more detail.

### Switching Conversational Styles

The use of different conversational styles when reading books is not uncommon. Fairy tales are read in a different style than a crime novel and reading of non-fiction differs even more from both. Different styles can also be found when reading a single text. In fictional texts, the most notable style switch occurs between narrative parts of the text and passages with direct speech. While narrative parts are normally rendered in a calm, flowing voice, direct speech has more varied prosody closer to spontaneous speech. This can be readily illustrated with the distribution of $F_0$ used by a speaker, as two examples for Librivox books in figure 7.1 show. The speaker of EMMA-II obviously uses the higher registers of $F_0$ much more frequently for direct speech. She is not simply shifting her voice to a higher tone but varying the melody more within her natural voice range. The same effect can be seen in the HAUFF reading albeit much less pronounced.

There has been some research on the automatic detection of voice styles in audio books, e.g. [SCCCB11] and [EBB12]. These approaches allow a more fine-grained clustering of speaking styles. However, while such detection is useful to train more specific prosody models, the problem remains to assign the right speaking style to the text during synthesis. In contrast, narrative parts can be easily identified from the text as discussed in 5.4.3.

To further evaluate the impact of these different styles on the prosody models, the audio books were split into narrative and direct speech sections. Two separate models were trained for the two styles. Then both models were validated against data from the same and the opposite style in order to understand how different the two models are. The Librivox books have been prepared as described in the previous two chapters. All models are trained using forward-only bottom-up training for the syntax-driven models. For each experiment, 6 different models were trained, taking out one chapter of the book as test material and using



(a) Hauff (German) (b) Emma (English)

Figure 7.1: *Normalized histogram of $F_0$ values for narrative and direct speech sections in the German* HAUFF *book and the English* EMMA-II *book.*

the remainder for the training of the prosody model. The numbers reported below are averages over these 6 trials. Details about the audio books used can be found in appendix B.

To validate the closeness of the models to natural speech, again an objective MSE validation of the normalized prosody parameters was employed. A short discussion of the different styles with respect to a subjective test will follow in the evaluation section 7.4 below. The results for the evaluation of the German HAUFF books against both the training corpus and the test material are listed in table 7.1. It should be noted that the MSE reported here is in general higher than for the models trained on the prosody corpora in chapter 4, more so for the direct speech style. This is to be expected as the readers of audio books have sufficient context for each sentence to be able to introduce an emotional and semantic component to prosody. Such context is not present in a corpus of isolated sentences and nor can it be learned by the syntax-driven model. This results in the better recall of the model for the prosody corpora but does not reflect how much of the functional prosody could actually be learned.

The results show that the different models can learn the specifics of their own style. The produced prosody is closer to the natural one of the same than the opposite style and, more importantly, this also holds for the most part for the test data. It is interesting to note that the

| training | narrative | | direct speech | |
|---|---|---|---|---|
| | dur (in ms) | $F_0$ (in Hz) | dur (in ms) | $F_0$ (in Hz) |
| all | 0.1960 | 0.1624 | 0.1963 | 0.1811 |
| narrative | 0.1944 | 0.1622 | 0.2041 | 0.1948 |
| direct speech | 0.1973 | 0.1671 | 0.1931 | 0.1722 |
| **recall** | **narrative** | | **direct speech** | |
| | dur (in ms) | $F_0$ (in Hz) | dur (in ms) | $F_0$ (in Hz) |
| all | 0.1961 | 0.1631 | 0.1984 | 0.1907 |
| narrative | 0.1964 | 0.1643 | 0.2053 | 0.1983 |
| direct speech | 0.1964 | 0.1671 | 0.1881 | 0.1885 |

Table 7.1: *MSE for $F_0$ and duration on German* HAUFF *text with a combined model using all training data and models separately trained with material for narrative and direct-speech style.*

$F_0$ model of the narrative style generalizes much better than for the direct speech model. Again, this reflects the fact that direct speech is generally realized with a more emotional voice, consequently containing more emphatic accents which cannot be learned from the syntax alone.

Compared to the general model trained over all training data, the improvement is still relatively low for the style-specific prosody models. For the test data, there is even a degradation visible for the narrative style. This is in part a shortcoming of the MSE evaluation method which favors a more average realization of prosody. Nonetheless, there is merit at this point to investigate which part of the prosody model is stronger affected by the style-specific training, the tree-derived part or the segmental part. To evaluate this two additional prosody models where trained where the statistic models are shared only partially between the styles. The *per-style syntax model* has a common segmental model but style-specific tree-derived models. The *per-style segmental model* accordingly shares only the tree-derived model. The results for the objective evaluation are again shown in table 7.2. There is little to no improvement with a per-style syntax model indicating that the syntax structure mainly relates to language-specific prosody and less to the individual style. The results for the pre-style segmental models vary for the different styles and prosody parameters. For the training material the full-style model always outperforms the one with combined

| training | narrative | | direct speech | |
|---|---|---|---|---|
| | dur | $F_0$ | dur | $F_0$ |
| combined | 0.1960 | 0.1624 | 0.1962 | 0.1811 |
| per-style syntax model | 0.1970 | 0.1642 | 0.1964 | 0.1781 |
| per-style segm. model | 0.1959 | 0.1645 | 0.1971 | 0.1818 |
| full per-style model | 0.1944 | 0.1622 | 0.1931 | 0.1722 |
| **recall** | **narrative** | | **direct speech** | |
| | dur | $F_0$ | dur | $F_0$ |
| combined | 0.1961 | 0.1631 | 0.1984 | 0.1907 |
| per-style syntax model | 0.1968 | 0.1650 | 0.1991 | 0.1903 |
| per-style segm. model | 0.1949 | 0.1646 | 0.2007 | 0.1892 |
| full per-style model | 0.1964 | 0.1642 | 0.1981 | 0.1885 |

Table 7.2: *MSE for $F_0$ and duration on German* HAUFF *text with different levels of separation for narrative and direct-speech style.*

tree-derived model but recall shows that this is partially due to a mild overfitting. One of the issues is that the direct speech parts of the text cover different syntax structures. The sentences in direct speech are on average shorter and less complex[1]. The training data for a combined tree-derived model is therefore more balanced with respect to syntactic coverage.

To evaluate if these findings are langauge-specific, we repeated the experiments on the EMMA-II book, the results of which are shown in table 7.3. The more varied prosody, already apparent in figure 7.1, is visible in these results as well. In particular, the direct speech style shows a higher MSE for $F_0$ indicating larger deviations from the mean speaking style but a lower variability in duration due to being spoken faster. Nonetheless, the relative differences between the models are similar to the HAUFF books.

[1] Detailed statistics over the coverage of the different styles of each book can be found in appendix B.

| training | narrative | | direct speech | |
|---|---|---|---|---|
| | dur | $F_0$ | dur | $F_0$ |
| combined | 0.1978 | 0.1908 | 0.1949 | 0.2396 |
| per-style syntax model | 0.1962 | 0.1838 | 0.1933 | 0.2380 |
| per-style segm. model | 0.1959 | 0.1827 | 0.1922 | 0.2380 |
| full per-style model | 0.1965 | 0.1827 | 0.1919 | 0.2370 |

| recall | narrative | | direct speech | |
|---|---|---|---|---|
| | dur | $F_0$ | dur | $F_0$ |
| combined | 0.1945 | 0.1978 | 0.1927 | 0.2478 |
| per-style syntax model | 0.1919 | 0.1918 | 0.1926 | 0.2499 |
| per-style segm. model | 0.1916 | 0.1901 | 0.1924 | 0.2458 |
| full per-style model | 0.1928 | 0.1860 | 0.1904 | 0.2490 |

Table 7.3: *MSE for $F_0$ and duration on English* EMMA-II *with different levels of separation for narrative and direct-speech style.*

## 7.2 Discourse Prosody

While the prosody style influences the outcome of the prosody generation as a whole, a text has also semantic and discourse-related properties that, like the syntactic features of the syntax tree, only have a locally constrained effect on the concrete prosody. These properties are much better included as additional features for the mutator functions.

In the following, some examples of such features will be discussed that can easily be extracted from the text. It will be outlined how they can be included in the prosody generation process. An implementation and evaluation of the features discussed in this section is subject for future work. Note that these properties are not specifically bound to the syntax-driven prosody generation. They can be integrated into any statistical prosody model. However, the hierarchic nature of the mutator model allows to bind these features to a specific scope. Thus, like synthetic features, they can be put into a larger context.

### 7.2.1 Surface Structure

In chapter 5, we have already discussed some of the aspects of the surface structure and how it helps to partition the text. This structure of chapters, paragraphs etc. should be preserved for the listener. The properties that can be extracted from the surface structure can be grouped into two classes:

**Sentence type** Headers, list, etc. can be represented as a special sentence whose type is not determined by its syntax structure but by how it is marked in the text. Like syntactic sentence types, it should be annotated as an additional attribute of the sentence constituent (the root node) in the syntax tree.[2]

**Sentence position** The surface structure can also help to put the sentence into a larger context as part of a paragraph. In a most simple way, this context can be described only through the position within the paragraph with special focus on the initial and final sentence of a paragraph. A more complex positioning should also take the surrounding sentences into account. At the moment the grammar only classifies the sentences into very coarse types (cf. 3.3.4). Taking into account the syntax structure of the sentence, a more fain grained distinction is possible that also allows to relate sentences with each other.

### 7.2.2 Discourse Structure beyond Syntax

The role of information flow for prosody has already been shortly discussed in chapter 3. We have seen that much of the relation between sentences can already be inferred from the syntax structure on one hand and the choice of function words on the other hand. Therefore, it can be expected that the information structure is already, at least partially, represented in the features for the mutator functions that were discussed in chapter 4. There is still a large group of properties of the discourse structure that are semantically and pragmatically motivated, for example contrast between nouns or special focus (cf. [HP86]). These constructs require to add semantic aspects to the analysis of the text.

A feature that is simple to introduce is the activation status of nouns, that is, the distinction between new and given information. It

[2] Special structures like headers or lists might also be modeled as their own style. The choice between style and feature is a practical one. In order for a surface structure to be modeled as a distinct style, sufficient training material needs to be available.

has been shown that the activation has an influence on accentuation, cf. [Bro83]. While tracking of the actual information content goes beyond simple text analysis, a simplified detection of givenness based on repetition of word stems is possible. This idea was discussed in [Hir93], where it is implemented as a stack of focus spaces which helped to improve the prediction of accents. For out prosody model, this would require additional features for each noun that state if the noun is new within the current scope, the current sentence and the current paragraph. If extended to functional words, the same mechanism can be used to detect some classes of repetitive structures like “They laughed about him, not about her.” which are also often assigned a special prosody.

This system could be further improved through detection of synonyms, hyponyms and hypernyms. [?] showed that they represent different types of accessibility resulting in different prosodic marking. Recent research into information structure also suggest that more fine-grained distinctions of the information status might provide a better description of focus [?]. However, the automatic inference of such categories remains a topic of active research (e.g. [?], [?]).

### 7.2.3 Emphatic Emphasis

The most difficult part of functional prosody is the inclusion of emphatic accents because it can often only be set correctly if the flow of the story as a whole is correctly understood. Therefore, the only solution in this case may be an annotation of the text with stage directions. Some authors employ a simplified form of such stage directions by using a special type setting for emphatic emphasis which can be detected as part of the text preprocessing.

Once detected, emphatic emphasis can be integrated in the prosody model through features of the mutator functions. The advantage over flat prosody models is here that the relation to the syntax tree allows to apply the emphasize feature to the exact part of the sentence to be emphasized. Emphasis on single words can be taken into account in the same way as an emphasis on complete phrases.

## 7.3 Mixed-lingual Texts

Another very common phenomenon in texts are foreign inclusions. They may appear in form of proper names, as movie or newspaper titles or there may even be entire sentences in a foreign language. The PolySVOX synthesis has the capability to parse multiple languages simultaneously and detect such foreign inclusions as part of the syntax analysis. This parsing is realized by having separate grammars for each language and very small inclusion grammars that describe how language switches can happen. The result are syntax trees where one or more subtrees are in a different language than the root of the tree. Further prosodic processing must thus be aware of the multiple languages. PolySVOX follows a mono-lingual approach where one language is chosen for each step and processing is done in that language. Phonological processing of the tree uses the language of the sentence, while subsequent processing uses the language of the syllable as a base, switching the language of the prosody model when the language of the syllable changes.

### 7.3.1 Foreign Inclusions by Human Speakers

The complete switch of prosody as done by PolySVOX simplifies the architecture but does not reflect what can be observed with human speakers. Humans tend to produce phonology and prosody that contains elements of base language and inclusion language. In fact, speakers seem to be confronted with two opposite goals. On the one side, they try to remain as close as possible to the realization of the embedded language. On the other hand, the inclusion must not disturb the flow of speech in the base language and be understandable even for listeners that do not speak the embedded language. This leads to a kind of stereotypic embedded language. It contains the typical elements that make it identifiable and understandable as a foreign language but is also heavily accented towards the base language. How much characteristic is retained depends on the language combination. While [OBK06] found that English inclusions in German sentences were preferred by the listener when they retained the English prosody, [ZT08] found the opposite for English inclusions in Chinese sentences.

### 7.3.2 Language Switching for Inclusions

The hierarchic model of prosody generation allows a true composition of prosody in mixed-lingual sentences while still using mono-lingual models trained on mono-lingual data. Given two mono-lingual syntax-driven prosody models, the tree prosody can be computed step-wise by going through the tree as explained in chapter 4. At each step, the mutator function of the appropriate language model is chosen, thus mixing both languages. As the overall target vector—the word prosody contour—is speaker-independent, models from different speakers can be mixed so that it is not necessary to have training material in multiple language from the same speaker. This is especially important for the syntax-driven model as more training material is required than for conventional models.

Using this model, the extent of influence of the embedded language depends on the length of the inclusion. Single words or noun phrases retain most of the prosody of the base language while complete phrases or partial sentences contain more of the embedded prosody. This ensures continuity in the prosody rendering and is an effect that could also be observed by the author in a study on language inclusions in natural speech [Hof10].

For segmental models, a binary choice has to be made between the segmental ANNs of base and embedded language. Both are possible and have different effects. Remaining in the base language might be better as the base rhythm is preserved, something observed in French and Chinese. Using the embedded language ANN might be better for understandability. In any case, as the segmental ANNs are speaker dependent, switching to the model of a different speaker might have discontinuity effects and therefore remaining in the base language may be the only option. Thus, while mixed-lingual support as outlined above has been implemented, more extensive listening tests will be necessary to evaluate the acceptability of such a system.

## 7.4 Perceptual Evaluation

The ultimate goal of an improved prosody generation is to achieve more natural sounding speech. This goal is not easy to evaluate because naturalness is not easy to define. There exists no accepted objective evaluation and if human subjects are asked to judge naturalness of speech, they are more likely to tie naturalness to the segmental quality of synthetic speech or to the acceptability of the voice itself (c.f [?]). In particular this last point makes a direct evaluation between TTS systems highly biased. This evaluation will therefore concentrate on evaluating how the syntax-driven prosody generation improves the prosody of a sentence compared to conventional accent/phrase-based prosody generation.

### 7.4.1 Evaluation Method

Different methods have been used to evaluate prosody. The PURR (Prosody Unveiling through Restricted Representation) method proposed in [SP98] reduces the stimuli to their prosodic content to ensure that any segmental influence is removed. This, however, would not allow to judge prosody with respect to the syntax of the sentence. We explicitly require that subjects can link the content of the sentence and its prosody. A more objective method was proposed in [WCM10]. It consists of an annotation test asking listeners to mark wrong boundaries and accents. This kind of evaluation is only applicable if there are true errors present in the concrete prosody. With the syntax-driven prosody, such errors are almost exclusively a result of errors in the parsing process, so that such a test would mainly evaluate the quality of the grammar.

To be able to concentrate on the naturalness of the prosody only, we use a modified version of the evaluation method used for SVOX [Tra95] and PolySVOX [Rom09a]. They applied a decision test where subjects were presented with sentences with natural and synthesized prosody in isolation and had to decide whether the sentence is natural or not. The main issue with this kind of evaluation is that subjects have to make an absolute decision without guidance what constitutes naturalness of a sentence. To make the task easier for non-expert listeners, the evaluation has been designed as a comparative task following the idea of controlled degradation as presented in [MJ03]. The subject is presented three different prosodic renderings of the same sentence and asked to order them by preference. The following prosody models have been used to create the sentences:

**Syntax-driven prosody (TREE)** The syntax-driven prosody model with the configuration described in chapter 4.

**Natural prosody (NAT)** The original prosody as applied by the speaker of the example sentence. It is considered as the best fitting prosody in the given context.

**Abstract prosody (FLAT)** As a comparison a model is trained that uses the phonological transcription as obtained by the PolySVOX system. The syntax analysis and the segmental model are the same as for the TREE model, so that the only difference is the way the syntax information is translated into prosody. This model uses only accentuation and phrasing derived with the rule-based models. It can be considered as producing the minimal, neutral prosody of the sentence and thus representing a lower bound of what can be achieved with a accent/phrasing model.

It should be emphasized at this point that the FLAT prosody does not represent the PolySVOX system of [Rom09a] which uses much more complex ANN models to produce prosody. A direct comparison between the two systems would be mute because the grammars of PolySVOX and the system presented in this thesis have a vastly differing coverage. A comparison is therefore more likely to result in a evaluation of this coverage than of the generated prosody.

To minimize the influence of segmental factors on the decision of the subjects, the prosodic parameters for all three models were computed and then implanted on the natural spoken sentence using LPC analysis-synthesis. This includes the natural prosody, where the sentence was resynthesized with the prosody parameters as they were extracted for the prosody training. The $F_0$ output was scaled so that all sentences used approximately the same $F_0$ voice range.

### Experimental Setup

The subjects were told that the experiment is about rating prosody and the instructions emphasized that they should concentrate on melody and rhythm. They were not told that one of the sentence represents human speech in order to not produce a bias.

The focus of the experiments were prosody models derived from audio books. We trained 6 different prosody models, three German models (Librivox books HAUFF, KELLER and DUNCKER) and three English models (Librivox books EMMA-I, EMMA-II, EMMA-III). A differentiation of style was made for the segmental models as discussed in the previous section. From each book 30 sentences were chosen that were not part of the training material. Sentences had a maximum length of 10 seconds. This excludes more complex sentence structures from the test but preliminary trials of the test setup showed that a useful comparison can only be made if the listener can retain the complete sentence in memory. Sentences with parse errors and foreign inclusions were excluded.

Participants were asked to rate their language proficiency before starting the text. Given the difficulty of the task, they were only presented with sentences in a language where they rated themselves as fluent or native speakers. In total, they had to rate 30 sentences per language, 10 sentences per voice. Within each language the sentences of the speakers were presented in random order so that the listener would not get overly used to a particular voice.

The experiment was accessible via an Internet page[3] and participation was done remotely. Subject could pause the survey at any time and return and finish it later.

### 7.4.2 Evaluation Results

**Subjects**

In total 24 people volunteered to participate in the evaluation. 6 had some form of speech training (as speech researcher, linguist or actor), the others were naive subjects. The German part of the survey was completed by 20 subjects, all of which were native speakers of German or Swiss German. The English part was completed by 19 people in total, only 3 of them were native speakers.

[3] The survey is still available at `http://speech.lonvia.de/en/survey`. In addition, a listing of all sentences used in the survey has been added after the survey had been closed on a separate page at `http://speech.lonvia.de/en/signals`.

| | all | | | naive | | | non-naive | | |
|---|---|---|---|---|---|---|---|---|---|
| rank | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| NAT | 83.0 | 9.4 | 7.6 | 82.9 | 9.6 | 7.5 | 84.0 | 8.0 | 8.0 |
| TREE | 10.9 | 52.8 | 36.3 | 10.9 | 52.4 | 36.7 | 10.5 | 55.0 | 34.5 |
| FLAT | 6.1 | 37.8 | 56.1 | 6.2 | 38.0 | 55.8 | 5.5 | 37.0 | 57.5 |

Table 7.4: *Overall preference ranking of the different prosody models: percentage of sentences ranked best (1) to worst (3). Results are for all listeners and divided by naive and non-naive listeners.*

### Results

Table 7.4 shows the ranking assigned over all evaluation data. Despite the subjects not knowing about the origin of the prosodies, the natural contour could still be identified with a relatively high accuracy and was preferred for 83 % of the sentences. All subjects described the task as challenging, although non-naive subjects reported that they were better able to distinguish small differences in the prosody. They indicated that they clearly heard three different levels of prosody while naive subjects found it more difficult to identify any difference between TREE and FLAT prosody. This is also reflected in the number of times they listened to the speech sample. While the sentence with natural prosody was played 1.7 times on average, the one with TREE prosody was played on average 2.0 times. Despite that there is a high agreement between the overall rating of naive and non-naive subjects, with non-naive subjects showing only a slightly stronger preference towards NAT and TREE prosodies. This indicates that despite the perceived difficulty, both groups were able to give consistent statements about their preferences.

When breaking down the results by book in figure 7.2, it can be seen that this preference varies greatly by book, which in fact is a direct reflection on the consistency the readers of these books showed in terms of prosody. The two books with the highest preference for the natural sentence, HAUFF and EMMA-I, also had the most consistent prosody with a calm speaking style and relatively subtle variances in their prosodic styling. EMMA-III employed a mix of American and British English which may have had a negative effect on the prosody as

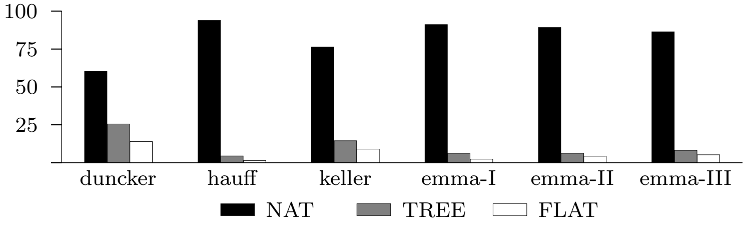


Figure 7.2: *Distribution of best rated utterance broken down by audio book.*

well. Finally, the speakers of the two other German books occasionally used a non-standard prosody style, for example by raising the voice at the end of statement sentences. Consequently, the more consistent synthetic prosody was frequently preferred.

Disregarding the natural sentences, figure 7.3 compares the preference between TREE and FLAT prosody only. Overall, for about 60 % of the sentences the syntax-driven prosody has been preferred indicating that the model can express prosody in a way that goes noticeably beyond a simple neutral accent and phrase structure. Again large differences can be observed between the different books. Most notable are the differences between the English books. While for EMMA-I there was a clear preference for the TREE prosody across all listeners, the other two books only have a very minor although still significant preference. As all three models are based on the reading of the same book, the difference cannot be attributed to differences in coverage of the training material and must therefore be related to the specific reading style of the speaker. Breaking the results further down by speaking style (see figure 7.4), the preference of the TREE prosody for the narrative style correlates well with the preference of the natural prosody, so that the overall pleasantness of the voice can explain the differences. The prosody for the sentences from direct speech, however, exhibits a different pattern. The direct speech parts of the EMMA-I books have received by far the highest preference for the TREE prosody. This difference can most likely be attributed to the consistent and less emotional speaking style of the reader, especially as the same preference of direct speech sentences can be seen for the HAUFF book. For one part, this shows

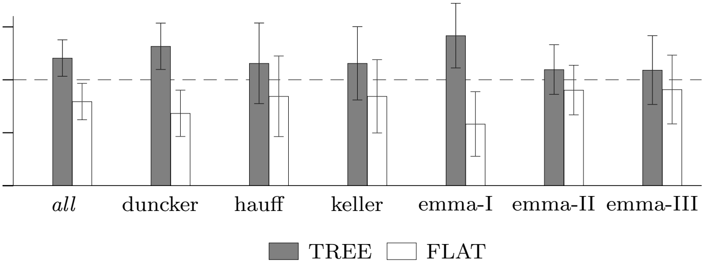


Figure 7.3: *Preference between TREE and FLAT prosody by book. The error bars mark the standard deviation between the individual participants.*

that the prosody model can learn the peculiarities of speakers, but for the other, it underlines the importance of consistency in the training material.

The German books use different texts as a base, so that the preference for one or another prosody model may be influenced at least partially by the coverage of the training material. Nonetheless it is interesting to note that for the DUNCKER book, which had the lowest preference for the natural prosody, subjects showed a much higher preference for the TREE prosody than for the other two books, with little difference between the two speaking styles. It seems that the syntax-driven prosody model is better able to cope with the non-standard prosody of the speaker than the FLAT prosody model during training. The same over-generalization that reduces the performance of the models for the more lively direct speech style works here in favor to produce more consistent models.

Finally, figure 7.5 shows the preference of the syntax-driven prosody model by sentence type. It can be expected that sentences other than statements have higher preference for the TREE prosody because the FLAT prosody model includes only statement phrase types (progressive and terminal phrase). The results confirm this assumption with the exception of German question sentences, which is the only sentence type for which the FLAT prosody was slightly preferred. A possible

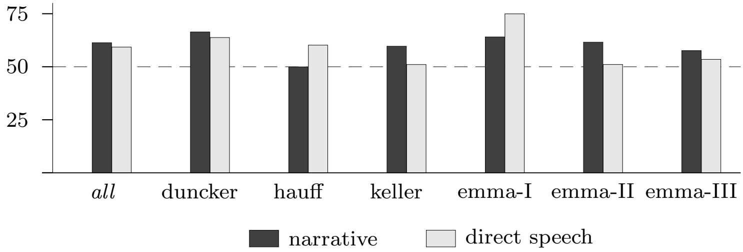


Figure 7.4: *Comparison of prosody preference between narrative and direct speech style by book. The bars correspond to the percentage of sentences where TREE prosody was prefered over FLAT prosody.*

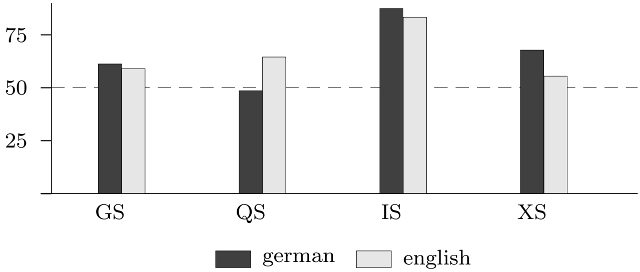


Figure 7.5: *Preference of TREE prosody over FLAT prosody by sentence type: statement (GS), question (QS), imperative (IS) and exclamation (XS).*

reason for this is that the German grammar does not sufficiently distinguish between the different question types. Or to be more precise, the difference cannot be learned because the locality principle of the mutator functions prevents a sufficiently large context to be taken into account to differentiate these types. Such issues can likely be corrected by further adapting the grammar and introducing more question types.

### 7.4.3 Discussion

In this chapter we have applied the syntax-driven prosody generation to audio books, using automatically preprocessed audio books to train models from different speakers and in two different languages.

The evaluation has shown that the prosody produced by the model is preferred with approximately 60 % over a more simple accent/phrase-based model. This indicates that not only the accentuation and phrasing structure can be learned directly from the syntax tree but that it also represents a more detailed variation of prosody. However, the evaluation has also shown that syntax alone is not sufficient to reproduce the full range of human prosody variations. In the beginning of this chapter, we have outlined some possible solutions involving a limited semantic analysis or an explicit annotation of the text. The syntax-driven prosody model does not preclude such annotations. On the contrary, the compositional nature of the model allows to specify much more precisely the scope of such annotations. As with the relation between syntax and prosody, this allows to think about these prosodic factors on a more functional level without having to consider the concrete realization.

The subjective evaluation in this chapter was still restricted to isolated sentences, so that the subjects had relatively little information about the context the sentence appeared in. For future work, an evaluation of the prosody synthesis for longer texts will need to be done to gain a better understanding how much of the discourse structure can be captured by the syntax-driven model and what further information the model requires to get even closer to the speaking performance of a human.

# Chapter 8

# Conclusion

## 8.1 Discussion

In this thesis, we have presented a statistical method for direct prosody generation from the results of the syntactic analysis of a text. We have shown that a compositional approach to prosody generation allows the inclusion of more complex syntactic relations than conventionally used in state-of-the-art statistical prosody generation approaches today. The central element to a more lively prosody is the availability of a grammar that models those syntactic constructs that are relevant for the realization of prosody. The more varying constructs are covered, the more nuances can be learned and reproduced. In this aspect, the approach is similar to rule-based prosody generation methods. However, as the model is able to learn the relation between syntax and prosody, it has the advantage that the constructs can be formulated in terms of syntactic rules, it is not necessary to understand how prosody will be realized in the end or, indeed, if the syntactic rule has an influence on the prosody at all.

The prosody model presented in this work does not rely on an abstract prosody layer as an intermediate stage between syntax and concrete prosody. It only uses a stylized prosody to separate segmental effects on prosodic parameters from actual functional prosodic effects. For the training of the model, this stylization can be directly computed from the signal properties of the natural speech, so that no further manual annotation of the training material is necessary. We have outlined a process that allows to prepare new training material automatically and without dependencies on externally trained models to predict such abstract parameters and applied this to different audio books. For these books the only manual preparation work necessary was to review the transcription to ensure that the formatting is compatible with the preprocessing step and to extend the dictionary with proper names used in the book to improve the quality of the parse step. Once done, further processing was completely automatic.

Finally, the prosody model allows in principle to incorporate functional prosodic properties into the prosody generation process that are not directly related to syntax. These properties still can profit from the structure established by the syntax tree to explicitly model relations between these features or restrict them to a well-defined context.

## 8.2 Outlook

The main challenge for automatic creation of prosodic models remains the noisy and very sparse training data. The random forest models used in this thesis could obtain satisfactory results, however, recent advances in machine learning can further help to improve the ability to learn prosodic structures. In particular, deep neural networks [HDY$^+$12] have shown promising results for speech recognition and might also be used for the mutator functions of the tree-derived model.

Another important factor for the acceptability of the prosody is the robustness of parsing. By coupling the prosody more tightly to the syntax structure of the sentence, errors in parsing are also more often noticeable in the prosody output. Therefore the grammars need to cover the language as completely as possible. In order to be rapidly transferable between languages, a faster mode of developing grammars needs to be further explored. Automatically learned grammars are an interesting option (e.g., [OG92], [HMBJ12]). Considering that parse errors can be detected by a human through the synthesized prosody, an automatic feedback between prosody production and grammar improvement might be possible as well.

This thesis has only touched on syntactic influences on prosody which still remains an important but relatively small part of possible influence factors. Including semantic information into the prosody generation will be required in order to produce truly natural prosody. The methods introduced in this thesis may well be applicable for a statistical learning of the resulting prosody, however, the main issue here remains to find reliable and fast methods to extract the information from the text.

In this context, concept-to-speech systems provide an interesting application for this kind of prosody generation. They are for example used in dialog systems and have the advantage over TTS systems that the intent behind the synthesized sentence is well known. That allows to generate features regarding information structure as discussed in 7.2 directly as part of the natural language generation process. There are already systems that focus on easier comprehensible discourse structure (e.g. [**?**]) and studies have shown that the inclusion of prosody can improve acceptability further (e.g. [**?**]). The challenge with creating syntax-derived prosody models for such systems lies in the creation of appropriate training material. However, the syntactic structures that such a system can produce are more deterministic and generally very well known in advance. As the training material only needs to be exhaustive with respect to syntactic coverage but not lexical coverage, a smaller recorded corpus might be sufficient.

One of the points that was not yet touched by this thesis is an investigation what parts of the generation contribute to language-specific prosody effects and which are speaker- and style-specific. If models can be better separated, new styles can be learned with much less training material.

# Appendix A

# Segmental Neural Networks

The segmental model of the second stage of prosody generation (see 4.3) has been realised with three independent ANNs. This appendix lists the configuration for each of these networks and the segmental input features.

Two different network configurations have been used. The duration ANN consists of simple 2-layer perceptron with one hidden layer. The $F_0$ and pause ANN models are recurrent networks of Elman type [Elm90] as shown in figure A.1. They have two hidden layers where the second layer is used as feedback for the next invocation of the net. The input is linearly normalized to a $[-1, 1]$ interval, the nodes in the hidden and output layers of the network use a sigmoid transfer function.

## A.1 $F_0$ Model

The $F_0$ ANN is a recurrent network. Its output represents 5 $F_0$ points per syllable: beginning of syllable onset and core, center of syllable nucleus, end of syllable nucleus and coda.

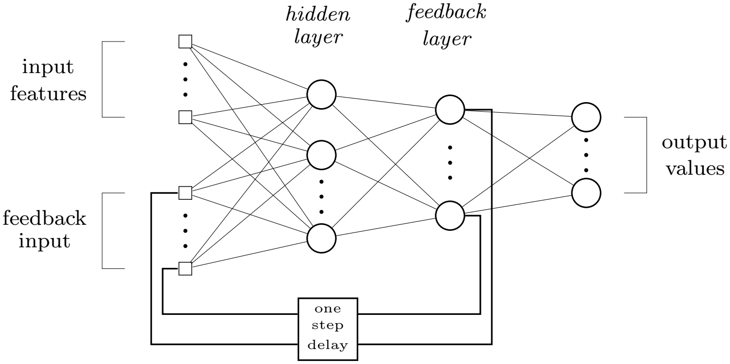


Figure A.1: *Network configuration of recurrent ANNs. The network has two hidden layers where the second is fed back to the input layer. The feedback inputs are initialized with zero at the beginning of each sentence.*

## Configuration

| | |
|---|---|
| Input features: | 30 + 7 word prosody contour features |
| Hidden layer: | 25 nodes |
| Feedback layer: | 10 nodes |
| Output layer: | 5 nodes |

## Input Features

| Scope | Feature | Context |
|---|---|---|
| syllable nucleus | length (short/medium/long)<br>high pitch vowel<br>1st format of nucleus (low/mid/high)<br>contains syllabic consonant<br>contains diphthong | +/- 1 |
| syllable onset | contains plosive<br>syllable has onset<br>contains voiced consonant | +1 |

| Scope | Feature | Context |
|---|---|---|
| syllable coda | contains plosive | |
| | syllable has coda | -1 |
| | contains voiced consonant | |
| | consonant position (front/mid/back) | +/-1 |
| foot | position | – |
| | length | |
| | salient | |
| word | syllable position | – |
| | length | |
| | first syllable | |
| | last syllable | |
| | before/at/after main stress | |
| | position relative to main stress | |
| syllable | length | +/- 1 |
| | stressed | |
| | primary stress | |

## A.2 Duration Model

Duration is determined for each phone separately.

### Configuration

| | |
|---|---|
| Input features: | 68 + 7 word prosody contour features |
| Hidden layer: | 30 nodes |
| Output layer: | 1 node |

### Input Features

| Scope | Feature | Context |
|---|---|---|
| phone | short segment<br>long segment<br>position (front/mid/back)<br>voicing<br>plosive<br>trill<br>fricative<br>pre-plosvie pause | +/-2 |
| syllable | position (onset/nucleus/coda)<br>length<br>stressed | – |
| syllable nucleus | long nucleus<br>high pitch nucleus<br>1st format of nucleus (low/mid/high)<br>contains syllabic consonant<br>contains diphthong | +/- 1 |
| syllable onset | long onset<br>contains glottal stop<br>contains plosive<br>syllable has onset<br>contains voiced consonant<br>contains fricative | +1 |

| Scope | Feature | Context |
|---|---|---|
| syllable coda | long coda<br>contains glottal stop<br>contains plosive<br>syllable has coda<br>contains voiced consonant<br>contains fricative | -1 |
| word | syllable position<br>first syllable<br>last syllable<br>before/at/after main stress<br>position relative to main stress | – |
| foot | position<br>salience | – |

## A.3 Pause Model

The pause ANN is also a recurrent network. It receives two word prosody contours, one for the word before the pause and one for the word that follows. The output is the normalized pause length. A pause is only realized if the computed length is at least 50 ms.

### Configuration

| | |
|---|---|
| Input features: | 52 + 2 x 7 word prosody contour features |
| Hidden layer: | 30 nodes |
| Feedback layer: | 3 nodes |
| Output layer: | 1 node |

### Input Features

| Scope | Feature | Context |
|---|---|---|
| phone | short segment<br>long segment<br>voicing<br>plosive<br>glottal stop<br>pre-plosvie pause | +/-3 |
| syllable | length<br>stressed | +/- 2 |
| word | length | +/- 1 |

# Appendix B

# Librivox Audio Books

This appendix lists all audio books used for experiments in this thesis. The books are taken from the Librivox project and are available at `http://librivox.org` together with the transcripts.

## B.1 Books for Prosody Models

The following books have been used for creating prosody models in chapter 7. Next to the source, statistics over sentence coverage by style are listed.

### Hauff (German)

| | |
|---|---|
| **Book:** | Märchen Almanach auf das Jahr 1826, 1827, 1828 |
| **Author:** | Wilhelm Hauff |
| **Speaker:** | Hokuspokus (female) |
| **Chapters:** | 39 |

| style | sentences | words/sent. | tree depth |
|---|---:|---:|---:|
| narrative | 3384 | 28.95 | 8.39 |
| direct speech | 2418 | 20.62 | 7.41 |

## Keller (German)

| | |
|---|---|
| **Book:** | Ferien vom Ich |
| **Author:** | Paul Keller |
| **Speaker:** | Rebecca Braunert-Plunkett (female) |
| **Chapters:** | 16 |

| style | sentences | words/sent. | tree depth |
|---|---|---|---|
| narrative | 3373 | 17.32 | 7.59 |
| direct speech | 2722 | 12.65 | 6.77 |

## Duncker (German)

| | |
|---|---|
| **Book:** | Großstadt |
| **Author:** | Dora Duncker |
| **Speaker:** | Jessi (female) |
| **Chapters:** | 21 |

| style | sentences | words/sent. | tree depth |
|---|---|---|---|
| narrative | 3428 | 18.49 | 7.91 |
| direct speech | 1596 | 13.19 | 6.86 |

## Emma-I (English)

| | |
|---|---|
| **Book:** | Emma |
| **Author:** | Jane Austen |
| **Speaker:** | Sherry Crowther (female) |
| **Chapters:** | 55 |

| style | sentences | words/sent. | tree depth |
|---|---|---|---|
| narrative | 3566 | 25.79 | 9.84 |
| direct speech | 5749 | 16.39 | 8.18 |

### Emma-II (English)

| | |
|---|---|
| **Book:** | Emma (Version 3) |
| **Author:** | Jane Austen |
| **Speaker:** | Elizabeth Klett (female) |
| **Chapters:** | 55 |

| style | sentences | words/sent. | tree depth |
|---|---|---|---|
| narrative | 3566 | 25.79 | 9.84 |
| direct speech | 5749 | 16.39 | 8.18 |

### Emma-III (English)

| | |
|---|---|
| **Book:** | Emma (Version 5) |
| **Author:** | Jane Austen |
| **Speaker:** | Moira Fogarty (female) |
| **Chapters:** | 55 |

| style | sentences | words/sent. | tree depth |
|---|---|---|---|
| narrative | 3566 | 25.79 | 9.84 |
| direct speech | 5749 | 16.39 | 8.18 |

## B.2 Books for Segmentation

The following aditional books have been used for evaluating language-independence of the segmentation process.

### Miserables (French)

| | |
|---|---|
| **Book:** | Les Miserables - tome 1 |
| **Author:** | Victor Hugo |
| **Speaker:** | Didier (male) |
| **Chapters:** | 70 |

## Helsinkiin (Finish)

| | |
|---|---|
| **Book:** | Helsinkiin |
| **Author:** | Juhani Aho |
| **Speaker:** | Tuija Aalto (female) |
| **Chapters:** | 6 |

## Legendi (Bulgarian)

| | |
|---|---|
| **Book:** | Staroplaninski legendi |
| **Author:** | Yordan Yovkov |
| **Speaker:** | Euthymius (male) |
| **Chapters:** | 10 |

# Bibliography

[ABGC05] J. Adell, A. Bonafonte, J.A. Gomez, and M.J. Castro. Comparative study of automatic phone segmentation methods for TTS. In *Proceedings of ICASSP*, pages 309–312, 2005.

[AEM01] G. Aversano, A. Esposito, and M. Marinaro. A new text independent method for phoneme segmentation. In *Proceedings of the 44th IEEE Midwest Symposium on Circuits and Systems*, pages 516–519, 2001.

[Alt87] B. Altenberg. Prosodic patterns in spoken English: Studies in the correlation between prosody and grammar for text-to-speech conversion. *Lund Studies in English*, 76, 1987.

[AOB11] G. K. Anumanchipalli, L. C. Oliveira, and A. W. Black. A statistical phrase/accent model for intonation modeling. *Proceedings of Interspeech*, pages 1813–1816, 2011.

[APL84] M. Anderson, J. Pierrehumbert, and M. Liberman. Synthesis by rule of english intonation patterns. In *Proceedings of ICAASP*, pages 77–80, 1984.

[Aus] J. Austin. *Emma.* Project Gutenberg, retrieved from `http://www.gutenberg.org/ebooks/158`.

[BCLM$^+$12] O. Boeffard, L. Charonnat, S. Le Maguer, D. Lolive, and G. Vidal. Towards fully automatic annotation of audiobooks for TTS. In *LREC*, pages 975–980, 2012.

[BF90] J. Bachenko and E. Fitzpatrick. A computational grammar of discourse-neutral prosodic phrasing in english. *Computational linguistics*, 16:155–170, 1990.

[BFO93] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on hidden markov models. *Speech Communication*, 12(4):357–370, 1993.

[BFOS84] L Breiman, J. Friedman, J. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[BGB10] N. Braunschweiler, M.J.F. Gales, and S. Buchholz. Lightly supervised recognition for automatic alignment of large coherent speech recordings. In *Proceedings of Interspeech*, pages 2222–2225, 2010.

[BH05] G. Bailly and B. Holm. SFC: A trainable prosodic model. *Speech Communication*, 46:348–364, 2005.

[Bie66] M. Bierwisch. Regeln für die Intonation deutscher Sätze. *Studia grammatica*, 7:99–201, 1966.

[BL89] S. Billot and B. Lang. The structure of shared forests in ambiguous parsing. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, pages 143–151, 1989.

[Bla96] A. Black. Generating F0 contours from ToBI labels using linear regression. In *Proceedings of ICSLP*, pages 1385–1388, 1996.

[BN08] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50:434–451, 2008.

[Bol72] D. Bolinger. Accent is predictable (if you're a mind-reader). *Language*, pages 633–644, 1972.

[Bre01] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[Bro83] G. Brown. Prosodic structure and the given/new distinction. In *Prosody: Models and measurements*, pages 67–77. Springer, 1983.

[BSA$^+$10] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R. C. Rose, and Thomas S. Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models. In *Proceedings of ICASSP*, pages 4334–4337, 2010.

[Cha88] W. Chafe. Punctuation and the prosody of written language. *Written communication*, 5:395–426, 1988.

[Cha94] W. Chafe. *Discourse, consciousness, and time: The flow and displacement of conscious experience in speaking and writing.* University of Chicago Press, 1994.

[CM06] R. Carter and M. McMarthy. *Cambridge Grammar of English.* Cambridge University Press, 2006.

[DB05] L. C. Dilley and M. Brown. *The RaP (Rhythm and Pitch) Labeling System v. 1.0.* Massachusetts Institute of Technology, 2005.

[DBG$^+$06] L. Dilley, M. Breen, E. Gibson, M. Bolivar, and J. Kraemer. A comparison of inter-transcriber reliability for two systems of prosodic annotation: Rap (rhythm and pitch) and tobi (tones and break indices). In *Proceedings of Interspeech*, 2006.

[Dor96] C. Doran. Punctuation in quoted speech. *arXiv preprint cmp-lg/9608011*, 1996.

[Dud09] Dudenredaktion. *Duden - Die Grammatik.* Dudenverlag Mannheim-Wien-Zurich, 2009.

[EBB12] F. Eyben, S. Buchholz, and N. Braunschweiler. Unsupervised clustering of emotion and voice styles for expressive TTS. In *Proceedings of ICAASP*, pages 4009–4012, 2012.

[EHP09] T. Ewender, S. Hoffmann, and B. Pfister. Nearly perfect detection of continuous F0 contour and frame classification for TTS synthesis. In *Proceedings of Interspeech*, pages 100–103, 2009.

[Elm90] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[FB89] E. Fitzpatrick and J. Bachenko. Parsing for prosody: what a text-to-speech system needs from syntax. In *Proceedings of the Annual AI Systems in Government Conference*, pages 188–194, 1989.

[FO98] H. Fujisaki and S. Ohno. The use of a generative model of F0 contours for multilingual speech synthesis. In *Proceedings of 4th International Conference on Signal Processing*, pages 714–717, 1998.

[Fuj88] H. Fujisaki. A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour. *Vocal Fold Physiology: Voice Production, Mechanisms and Functions*, pages 347–355, 1988.

[GE72] F. Goldman-Eisler. Pauses, clauses, sentences. *Language and Speech*, 15(2):103–113, 1972.

[GKMN07] M. Grimm, K. Kroschel, E. Mower, and S. Narayanan. Primitives-based evaluation and estimation of emotions in speech. *Speech Communication*, 49(10):787–800, 2007.

[GLF$^+$93] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue. TIMIT acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium, Philadelphia*, 1993.

[GO07] L. Golipour and D. O'Shaughnessy. A new approach for phoneme segmentation of speech signals. In *Proceedings of Interspeech*, pages 1933–1936, 2007.

[HB96] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of ICASSP*, pages 373–376, 1996.

[HC98] D. Hirst and A. Di Cristo. *Intonation systems: A Survey of Twenty Languages*, chapter A survey of intonation systems. Cambridge University Press, 1998.

[HCB$^+$93] D. Hirst, A. Di Cristo, M. Le Besnerais, Z. Najim, P. Nicolas, and P. Romeas. Multi-lingual modelling of intonation patterns. In *Proceedings of ESCA Workshop on Prosody*, pages 204–207, 1993.

[HDY$^+$12] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing*, 29:82–97, 2012.

[Hir93] J. Hirschberg. Pitch accent in context predicting intonational prominence from text. *Artificial Intelligence*, 63(1-2):305–340, 1993.

[Hir99] D. Hirst. The symbolic coding of segmental duration and tonal alignment. an extension to the INTSINT system. In *Proceedings of Eurospeech*, 1999.

[Hir02] J. Hirschberg. Communication and prosody: Functional aspects of prosody. *Speech Communication*, 36:31–43, 2002.

[HK07] A. Haubold and J.R. Kender. Alignment of speech to highly imperfect text transcriptions. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 224 –227, july 2007.

[HMBJ12] D. Hrnčič, M. Mernik, B. R. Bryant, and F. Javed. A memetic grammar inference algorithm for language learning. *Applied Soft Computing*, 12(3):1006–1020, 2012.

[Hof10] S. Hoffmann. Preliminary study of prosody in foreign language inclusions. Technical report, ETH Zurich, 2010.

[HP86] J. Hirschberg and J. Pierrehumbert. The intonational structuring of discourse. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pages 136–144, 1986.

[HP10] S. Hoffmann and B. Pfister. Fully automatic segmentation for prosodic speech corpora. In *Proceedings of Interspeech*, pages 1389–1392, 2010.

[HP12] S. Hoffmann and B. Pfister. Employing sentence structure: Syntax trees as prosody generators. In *Proceedings of Interspeech*, 2012.

[HP13] S. Hoffmann and B. Pfister. Text-to-speech alignment of long recordings using universal phone models. In *Proceedings of Interspeech*, pages 1520–1524, 2013.

[HR01] J. Hirschberg and O. Rambow. Learning prosodic features using a tree representation. In *Proceedings of Eurospeech*, pages 1175–1178, 2001.

[Kay82] M. Kay. Algorithm schemata and data structures in syntactic processing. In *Text Processing: Text Analysis and Generation, Text Typology and Attribution*, pages 327–358. Almqvist and Wiksell International, Stockholm, 1982.

[KBG$^+$11] A. Katsamanis, M. P. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan. Sailalign: Robust long speech-text alignment. *Proceedings of Workshop on New Tools and Methods for Very Large Scale Research in Phonetic Sciences*, 2011.

[KC02] Y. J. Kim and A. Conkie. Automatic segmentation combining an HMM-based approach and spectral boundary correction. In *Proceedings of ICSLP*, pages 145–148, 2002.

[KCFH03] P. A. Keating, T. Cho, C. Fougeron, and C. S. Hsu. Domain-initial articulatory strengthening in four languages. *Papers in laboratory phonology VI*, pages 145–163, 2003.

[Ken12] G. Kentner. Linguistic rhythm guides parsing decisions in written sentence comprehension. *Cognition*, 123:1–20, 2012.

[Kip66] P. Kiparsky. Über den deutschen Akzent. *Studia Grammatica*, VII:69–98, 1966.

[KKG02] A. Koriat, H. Kreiner, and S. N. Greenberg. The extraction of structure during reading: Evidence from reading prosody. *Memory & cognition*, 30:270–280, 2002.

[Kos83] K. Koskenniemi. Two-level model for morphological analysis. In *IJCAI*, pages 683–685, 1983.

[KS03] G. Kochanski and C. Shih. Prosody modeling with soft templates. *Speech Communication*, 39(3):311–352, 2003.

[LDY$^{+}$09] Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. A study on multilingual acoustic modeling for large vocabulary asr. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009.

[LMG$^{+}$11] A. Lazaridis, I. Mporas, T. Ganchev, G. Kokkinakis, and N. Fakotakis. Improving phone duration modelling using support vector regression fusion. *Speech Communication*, 53:85–97, 2011.

[MA09] P. J. Moreno and C. Alberti. A factor automaton approach for the forced alignment of long speech recordings. In *Proceedings of ICASSP*, pages 4869–4872, 2009.

[MDM98] F. Malfrère, T. Dutoit, and P. Mertens. Fully automatic prosody generator for text-to-speech. In *5th International Conference on Spoken Language Processing*, pages 1395–1398, 1998.

[MJ01] H. Mixdorff and O. Jokisch. Building an integrated prosodic model of German. In *Proceedings of Interspeech*, pages 947–950, 2001.

[MJ03] H. Mixdorff and O. Jokisch. Evaluating the quality of an integrated model of german prosody. *International journal of speech technology*, 6:45–55, 2003.

[MJVTG98] P.J. Moreno, C. Joerg, J.M. Van Thong, and O. Glickman. A recursive algorithm for the forced alignment of very long audio segments. In *Proceedings of ICSLP*, 1998.

[MR03] J. R. Martin and D. Rose. *Working with discourse: Meaning beyond the clause.* Continuum International Publishing Group, 2003.

[OBK06] P. Olaszi, T. Burrows, and K. Knill. Investigating prosodic modifications for polyglot text-to-speech synthesis. In *Proceedings of MULTILING*, 2006.

[OG92] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. *Series in Machine Perception and Artificial Intelligence*, 1:49–61, 1992.

[OHH$^+$00] R. Ogden, S. Hawkins, J. House, M. Huckvale, J. Local, P. Carter, J. Dankovičová, and S. Heid. ProSynth: An integrated prosodic approach to device-independent, natural-sounding speech synthesis. *Computer Speech & Language*, 14:177–210, 2000.

[PB11] K. Prahallad and A. W. Black. Segmentation of monologues in audio books for building synthetic voices. *IEEE Transactions on Audio, Speech, and Language Processing*, 19:1444–1448, 2011.

[Pie80] J. B. Pierrehumbert. *The Phonology and Phonetics of English Intonation.* PhD thesis, Massachusetts Institute of Technology, 1980.

[PKW96] S. Pauws, Y. Kamp, and L. Willems. A hierarchical method of automatic speech segmentation for synthesis applications. *Speech Communication*, 19(3):207–220, 1996.

[POXT09] S. Prom-On, Y. Xu, and B. Thipakorn. Modeling tone and intonation in mandarin and english as a process of target approximation. *The Journal of the Acoustical Society of America*, 125:405–424, 2009.

[Rie95] M. Riedi. A neural-network-based model of segmental duration for speech synthesis. In *Proceedings of Eurospeech*, pages 599–602, 1995.

[Rie97] M. Riedi. Modeling segmental duration with multivariate adaptive regression splines. In *Proceedings of Eurospeech*, pages 2627–2630, 1997.

[Ril90] M. D. Riley. Tree-based modelling for speech synthesis. In *ESCA Workshop on Speech Synthesis*, pages 229–232, 1990.

[RNM99] F. Ramus, M. Nespor, and J. Mehler. Correlates of linguistic rhythm in the speech signal. *Cognition*, 73:265–292, 1999.

[RO99] K. N. Ross and M. Ostendorf. A dynamical system model for generating fundamental frequency for speech synthesis. *IEEE Transactions on Speech and Audio Processing*, 7:295–309, 1999.

[Rom09a] H. Romsdorfer. *Polyglot Text-to-Speech Synthesis: Text Analysis & Prosody Control.* PhD thesis, ETH Zurich, 2009.

[Rom09b] H. Romsdorfer. Weighted neural network ensemble models for speech prosody control. In *Proceedings of Interspeech*, pages 492–495, 2009.

[RP04] H. Romsdorfer and B. Pfister. Multi-context rules for phonological processing in polyglott TTS synthesis. In *Proceedings of Interspeech*, pages 737–740, 2004.

[RP06] H. Romsdorfer and B. Pfister. Character stream parsing of mixed-lingual text. In *Proceedings of MultiLing*, 2006.

[RP07] H. Romsdorfer and B. Pfister. Text analysis and language identification for polyglot text-to-speech synthesis. *Speech Communication (Elsevier)*, 49(9):697–724, 2007.

[RRM97] J. Robert-Ribes and R.G. Mukhtar. Automatic generation of hyperlinks between audio and transcript. In *Eurospeech*, 1997.

[SBP$^+$92] K. EA. Silverman, M. E. Beckman, J. F. Pitrelli, M. Ostendorf, C. W. Wightman, P. Price, J. B. Pierrehumbert, and J. Hirschberg. TOBI: a standard for labeling english prosody. In *Proceedings of ICSLP*, pages 867–870, 1992.

[SCCCB11] E. Székely, J. P. 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.

[Sel80] E. O. Selkirk. *On prosodic structure and its relation to syntactic structure.* Indiana University Linguistics Club, 1980.

[Sel86] E. O. Selkirk. On derived domains in sentence phonology. *Phonology yearbook*, 3:371–405, 1986.

[SHNG10] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj. Analysis of duration prediction accuracy in hmm-based speech synthesis. In *Proceedings of Speech Prosody*, 2010.

[Sil87] K. EA. Silverman. *The structure and processing of fundamental frequency contours.* PhD thesis, University of Cambridge, 1987.

[Sil93] K. EA. Silverman. On customizing prosody in speech synthesis: names and addresses as a case in point. In *Proceedings of the workshop on Human Language Technology*, pages 317–322. Association for Computational Linguistics, 1993.

[SM00] A. K. Syrdal and J. T. McGory. Inter-transcriber reliability of ToBI prosodic labeling. In *Proceedings of Interspeech*, pages 235–238, 2000.

[SOS12] T. Schlippe, S. Ochs, and T. Schulz. Grapheme-to-phoneme model generation for indo-european languages. In *Proceedings of ICASSP*, pages 4801–4804, 2012.

[SP98] G. P. Sonntag and T. Portele. PURR—a method for prosody evaluation and investigation. *Computer Speech & Language*, 12:437–451, 1998.

[STC+03] A. Serralheiro, I. Trancoso, D. Caseiro, T. Chambel, L. Carrico, and N. Guimarães. Towards a repository of digital talking books. In *Proceedings of Eurospeech*, pages 1605–1608, 2003.

[Str02] V. Strom. From text to speech without ToBI. In *Proceedings of ICSLP*, 2002.

[Tay94] P. Taylor. The rise/fall/connection model of intonation. *Speech Communication*, 15:169–186, 1994.

[Tay00] P. Taylor. Analysis and synthesis of intonation using the tilt model. *The Journal of the acoustical society of America*, 107:1697–1714, 2000.

[TB94] P. A. Taylor and A. W. Black. Synthesizing conversational intonation from a linguistically rich input. In *ISCA/IEEE Workshop on Speech Synthesis*, pages 175–178, 1994.

[TGG03] D.T. Toledano, L.A.H. Gomez, and L.V. Grande. Automatic phonetic segmentation. *IEEE Transactions on Speech and Audio Processing*, 11(6):617–625, 2003.

[Tra95] C. Traber. *SVOX: The Implementation of a Text-to-Speech System for German.* PhD thesis, ETH Zurich, March 1995.

[vS94] J. van Santen. Assignment of segmental duration in text-to-speech synthesis. *Computer Speech and Language*, 8:95–95, 1994.

[vSKKM05] J. van Santen, A. Kain, E. Klabbers, and T. Mishra. Synthesis of prosody using multi-level unit sequences. *Speech Communication*, 46:365 – 375, 2005.

[WCM10] M. White, R. AJ. Clark, and J. D. Moore. Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, 36:159–201, 2010.

[WH92] M. Q. Wang and J. Hirschberg. Automatic classification of intonational phrase boundaries. *Computer Speech & Language*, 6:175–196, 1992.

[WSHOP92] C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. J. Price. Segmental durations in the vicinity of prosodic phrase boundaries. *The Journal of the Acoustical Society of America*, 91:1707–1717, 1992.

[YKK08] J. Yamagishi, H. Kawai, and T. Kobayashi. Phone duration modeling using gradient tree boosting. *Speech Communication*, 50:405–415, 2008.

[YOMK03] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Modeling of various speaking styles and emotions for hmm-based speech synthesis. In *Proceedings of Interspeech*, pages 2461–2464, 2003.

[You94] S.J. Young. The HTK Hidden Markov Model Toolkit: Design and philosophy. *Entropic Cambridge Research Laboratory, Ltd*, 2:2–44, 1994.

[YTM$^+$99] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proceedings of Eurospeech*, 1999.

[ZT08] Y. Zhang and J. Tao. Prosody modification on mixed-language speech synthesis. In *Proceedings of ISCSLP*, 2008.

# Curriculum Vitae

| | |
|---|---|
| **1978** | Born in Suhl, Germany |
| **1983-1990** | Primary school in Suhl |
| **1990-1992** | Polytechnic highschool in Suhl |
| **1992-1997** | Gymnasium in Suhl (Abitur) |
| **1997-2003** | Studies in computer science at Technische Universität Dresden |
| **2003** | Diploma in computer science (Dipl. Inf.) |
| **2003-2007** | Research engineer for ST Microelectronics, Rousset, France |
| **2007-2014** | Research assistant and PhD student at the Speech Processing Group, Computer Engineering and Networks Laboratory, ETH Zürich |