Style-Specific Phrasing in Speech Synthesis

Alok Parlikar

CMU-LTI-13-012

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Thesis Committee
Alan W Black
Carnegie Mellon
Florian Metze
Carnegie Mellon
Ian Lane
Carnegie Mellon
Kishore Prahallad
IIIT Hyderabad, India

Submitted in
partial fulfillment
of the requirements for
the degree of Doctor of Philosophy
in Language and Information Technologies

Abstract

People pause between words and sentences when they speak. They pause to emphasize content, or to make an utterance more understandable, or just to take a breath. A speech synthesizer should also insert similar pauses to sound natural.

The process of inserting prosodic breaks in an utterance is called *Phrasing*. Phrasing is a crucial step during speech synthesis because other models of prosody depend on it. Phrasing also helps characterize styles of speech, and synthesizers must adapt their phrasing to different speaking styles.

This thesis presents a data-driven grammar-based approach that can be used to build style-specific phrasing models. We automatically label phrase breaks from speech data and use features over acoustic syntax in our modeling. Experimental results, both objective and subjective, show that these models are better than the prior state-of-art across various speaking styles.

This thesis presents a minimum error-rate training approach to improve the phrasing models by optimizing them directly towards the evaluation criterion: the F-measure. This framework also allows us to define a knob that can be used to vary the number of phrase breaks produced in an utterance. This can be useful when changing the speaking rate.

This thesis also discusses modeling not just the placement of phrase breaks, but also their duration. Corpus analysis shows that durations of breaks vary quite significantly between different styles, and we present methods with which this variation can be captured in a way that is perceptually better.

The presented phrasing methods can have a broader impact on intonation models and can enhance the intelligibility of the synthesis of machine translation output. These methods can also be extended to "low-resource" scenarios, such as when building voices for uncommon languages, or for languages that do not have a standardized orthography.

Acknowledgments

This thesis would not have been possible without the infinite support of my teachers, friends and family: together they set up the carrier wave over which I could modulate this work and present it to you.

I would like to thank my thesis advisor, Alan W Black, for steering my research and shaping this thesis. I would also like to express my gratitude to my thesis committee members: Florian Metze, Ian Lane, and Kishore Prahallad, for providing me with unique perspectives and feedback that has helped refine the ideas in this thesis. I am indebted to the various funding agencies that supported me and my research at CMU.

I am very grateful to Alan for all his mentorship and advice on research, coding, writing and presentation. Working with him has taught me very valuable lessons in balancing the theory in research with its practical applications. He has always been willing to give me the time and resources to explore platforms such as Android and Amazon Mechanical Turk, and this has been a lot of fun and a source of overall motivation. I am extremely fortunate to have had the opportunity to work with Alon Lavie and Stephan Vogel as my MS advisors. They gave me my Research-101, took personal interest in my research and taught me a lot about statistical machine translation. That has had a great impact, technical and otherwise, on this thesis. I also thank the professors at my undergraduate college in India, specially Dr. Gorachand Nandi and Dr. Sudip Sanyal, who encouraged me to work on research problems in language processing and provided me guidance and resources in my endeavors.

A big 'thank you' to all my friends – those in Pittsburgh, those on the internet, and the imaginary ones. Special thanks to Abhimanyu Lad and Aasish Pappu for the humor that helped me stay sane. Many thanks also to Gopala Krishna Anumanchipalli, Prasanna Muthukumar, Sunayana Sitaram, Sukhada Palkar, Udhyakumar Nallasamy and Jon Clark, for their technical contributions and feedback that helped shape this thesis.

Most of all, my family: they merit the instrumental, dative and ablative cases in relation to this thesis. Their endless love, support, encouragement; their enabling me to dream big and teaching how to chase those dreams, leaves me unable to come up with words to thank them; all I seem to come up with is a deep, grateful pause; an emotion-filled phrase break;

Contents

Chapter 1

Introduction

A story is told as much by silence as by speech.

Susan Griffin

THE ULTIMATE OBJECTIVE of a speech synthesizer is to pass the Turing test: the speech that it generates be indistinguishable from human speech. There are two broad hurdles for synthetic speech in passing this test: (i) Spectral similarity to a human voice, and (ii) Presentation in an appropriate style of speech. Significant research has been devoted to achieving acoustic naturalness in speech. The Blizzard Challenge (Black and Tokuda, 2005) and its results over the past several years have shown that synthetic voices can sound very natural, and sound like the target speaker. As synthetic voices sound more and more natural, people start becoming less forgiving about errors in speech synthesis. Natural sounding speech that is presented in an inappropriate style can annoy listeners, require increased listener effort, or even reduce intelligibility of content. Thus the problem of synthesizing with appropriate speaking style needs to be thoroughly addressed now.

1.1 Speaking Styles

The very notion of speaking style is a nebulous one (Hirschberg, 2000). Different people have their own general speaking style. But the same person could adopt different styles when reading passages from different genres of text. The speech style could also vary depending on the task at hand. For example, if somebody were to dictate a certain text from a play to a typist, their style could be different from if they were actually enacting the play.

Labov (1964, 1972) has suggested that speaking style changes as a function of the attention that the speaker pays to his discourse. Although the cause of this change is related to the social setting of a discussion, there are constant variations in the style. Joos (1968) and Zwicky (1972) suggested that styles vary based on the casualness of the speaker in the given setting. Eskenazi (1993) explored and related the work on speaking styles and provided a data-driven definition to "speaking styles". According to her, style reflects the action of the environment upon the speaker, and the speaker upon the environment. It is also a projection of the speakers' projection of themselves, their background, and is a setting of the type and tone of the conversation they wish to have. The style is a result of conscious and subconscious effort on their part, and is not always perceived in the same way as it was intended. Some styles are easier than others for some people to express, and for some to perceive.

There are several dimensions in which speaking styles vary. Picheny et al. (1986); Chen (1980); Browman and Goldstein (1990); Labov (1964); Bladon et al. (1987); Granström (1992); Krull (1989); Duez (1992); Lindblom (1990) have suggested that there are acoustico-phonetic variations in style, as evident by the manifestation of consonants and vowels. Picheny et al. (1985); Gimson (1989); Browman and Goldstein (1990); Picheny et al. (1986); Labov (1964); Eskenazi and Lacheret-Dujour (1991) have shown evidence for phonological changes as style varies. These include phoneme insertion, phonological reduction, consonant deletion, etc. Word stress, pronunciation, sentence structure, and intelligibility also varies with the choice of speaker's style. Bruce (1995); Ayers (1994); Barry (1995); Kohler (1995); Blaauw (1992); Touati (1995); Parlikar and Black (2011) have observed that under different speech contexts, the voice quality, timing, intonation contour, pitch accents and phrasing all vary in speech.

A truly style-specific speech synthesizer should be able to produce speech with all the variations that we find in natural speech. This thesis addresses variations in style along one of the prosodic dimensions: phrase breaks in speech.

1.2 Phrase Breaks

Silverman et al. (1992) suggests that phrase breaks can be classified into multiple levels. One of the most prominent and frequent levels is a pause. People pause every now and then when they speak. These pauses are sometimes physiological and sometimes reflect cognitive processes. Speech production is a complex motor activity involving many organs at a time, and physiological limitations make it impossible to produce speech continuously. Physiologically inevitable pauses regularly occur with the breathing cycle. Speech production is also a rhythmic activity where word groups are produced at particular rates, and this also contributes to regular placement of pauses. Further, individual physiological constraints, such as strength and capacity of lungs, muscular tone and articulatory rate affect the number of pauses. Goldman-Eisler (1961) suggests that pauses are an external reflection of some of the cognitive activities involved in speech production. This activity could be on the part of the speaker (thinking before delivering a clear message), or on part of the listener (giving listeners time to assimilate what was just spoken). Grosjean and Deschamps (1975) have shown that the number of pauses is affected by the complexity of a communicative task.

Phrase breaks in natural speech are important: they are physiologically essential, help emphasize the right content, and improve the intelligibility of speech. However, in synthetic speech, these breaks are even more important.

A typical speech synthesizer consists of a cascade of several modules. Given a text form of an utterance, the text processing module expands tokens into words, and makes choice about the pronunciation of words based on their context. Prosody is generated by a combination of multiple models: the phrasing model, the duration model and the intonation model. This is then combined with spectral prediction to produce actual waveform speech.

Phrasing module is typically the first among various prosodic models, and it makes decisions about where breaks should go in the speech to be generated. This information is used by the other models of prosody. Because a phrasing model lays out foundation of synthetic prosody, a style-specific phrasing model is essential to produce speech in the desired styles. This thesis addresses the important problem of building a style-specific phrasing model.

1.3 Data-Driven Phrasing

The main problem associated with building "stylized" phrasing models is the effort required in eliciting sufficient training data. Phrasing models are typically trained on text corpus that is hand-annotated with breaks. In order to train a style-specific phrasing model, one would have to obtain manual annotations of phrase breaks over each of the styles we would like to handle. Linguistic annotations can be expensive. The investment (time, effort) required in getting such annotations would be prohibitive of building style-specific models for every synthetic voice that we build.

Our proposal in this thesis is to use data-driven techniques to gather training data for phrasing models. We shall see that not only is such an approach feasible, but it is also extensible to low-resource scenarios, such as in building phrasing models for new languages.

1.4 Thesis Statement

Style-specific phrasing is useful, novel, and feasible with the help of data-driven techniques. It is perceived by people as being better. It helps improve other models of prosody. It can make the synthesis of machine translation output more intelligible. Methods proposed in the thesis are extensible to low-resource scenarios.

1.5 Thesis Contributions

The work in this thesis is set within the framework of the Festival (Black and Taylor, 1997) speech synthesis engine. The research and development contributions are:

- A Data-Driven Grammar-Based Phrasing Model. See Chapter 2 and (Parlikar and Black, 2011).
- A Minimum-Error-Rate training approach to Phrasing. See Chapter 3 and (Parlikar and Black, 2013).
- Analysis and Modeling of duration of phrase breaks. See Chapter 4 and (Parlikar and Black, 2012b).
- Methods for improved synthesis in the context of speech to speech translation. See Chapter 5 and (Parlikar et al., 2010).
- Extension of presented methods to apply them under low-resource conditions. See Chapter 6 and (Parlikar and Black, 2012a).
- TestVox: An open-source, web-based subjective evaluation framework. See Appendix A.

Chapter 2

Phrase Break Prediction

A pause in the wrong place, an intonation misunderstood, and a whole conversation went awry.

E. M. Forster
Passage to India

People pause between words when they speak. They pause to emphasize something, or to make their utterance more understandable, or quite often, just to take a breath. A speech synthesizer should also pause in a similar manner. Appropriate pauses can enhance the intelligibility of speech and make it sound more natural. The process of determining where a synthesizer should insert these pauses is called phrase break prediction, or phrasing. Since phrase breaks happen at word boundaries, phrasing can also be defined as classifying each of the word boundaries in text as being a break, or a non-break.

Phrasing is a crucial step during speech synthesis. It breaks long utterances into meaningful units of information, and often resolves ambiguities in text, thereby making it more understandable. More importantly, in a typical speech synthesizer, phrase breaks lay the foundation required by other models of prosody, such as accent prediction (Hirschberg, 1993; Ross and Ostendorf, 1996) and duration modeling (van Santen, 1994). Any errors made in the initial phrasing step may get compounded by the other models and result in synthetic speech that either sounds unnatural, or is difficult to understand.

Phrase breaks are diverse: there is no one-single correct way to phrase some given text. Different people may phrase the same text differently. Different genres of text may lead a speaker to adopt different phrasing styles. Different languages may have inherently different phrasing patterns.

Prahallad et al. (2010) have shown that prosodic phrase breaks are specific to a speaker. In fact, some speakers have an extremely unique phrasing style. Two striking examples are the politicians: Barack Obama (44th president of the United States), and Atal Bihari Vajpayee (10th prime minister of India). In an informal experiment, we took short audio clips from recordings of Obama's speech and de-lexicalized them by changing the spectra in all syllables to correspond to the sound 'ma'. This transformed speech was played to a class of undergrads at Carnegie Mellon, and students were asked to identify the speaker. The students were able to unanimously identify the correct speaker, and the general consensus of their feedback was that they could identify it primarily because of the phrasing patterns. Figure 2.1 shows the phrasing profile of two speakers when reading the same text. This text was the output of a Chinese-English machine translation system. This plot shows the probability of different phrase lengths, i.e., probability of *x* words between two consecutive pauses. The profiles of the two speakers are quite different, thereby supporting the hypothesis that phrase breaks are speaker specific.

The style of phrase breaks can also depend on the domain of the text at hand. For example, a speaker may phrase differently when reading a book, compared to when reading a news article. Example of this phenomenon can be seen in Figure 2.2. The phrasing profiles compared here are analyzed from the same speaker, recording two different types of text. In one case the recording is of isolated sentences taken from the ARCTIC prompts. In the other case, the recording is of isolated sentences taken from the transcripts of EuroParl proceedings. We can see that the phrasing profiles for different types of text, even with the same speaker, are quite

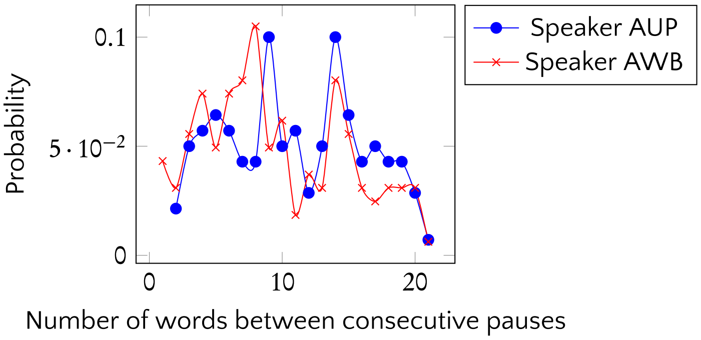


Figure 2.1: Phrasing profiles of two speakers reading the same text. Evidence that phrasing is speaker specific.

different.

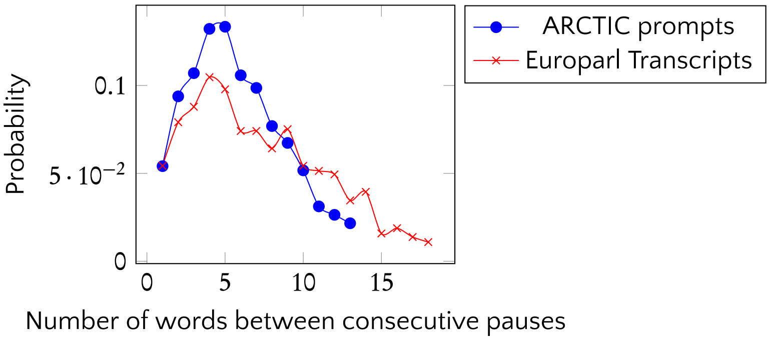


Figure 2.2: Phrasing profiles of one speaker (AUP) reading text in two genres. Evidence that phrasing depends on genres.

Kim and Oh (1996) have noted that faster speech tends to have fewer phrase breaks. Frota and Marina (2007) have suggested that language-particular preferences also influence phrasing patterns. All this evidence leads us to infer that phrasing styles are different in different situations.

Speech synthesizers need to adapt their phrasing model to specific speakers and speaking styles. Typical synthesizers today, such as the Festival speech synthesis system (Black and Taylor, 1997) use a generic phrasing model trained on one particular corpus of hand-annotated breaks.

The style of this phrasing model is not always appropriate for the voices we build. In situations where this generic phrasing model is inappropriate, intonational models such as (Anumanchipalli et al., 2011) are unable to appropriately capture stylistic prosody. We thus need a method to build phrasing models targeted towards the voice that we intend to build.

Phrasing models have typically been built from large corpora of manually annotated breaks. If we want a targeted phrasing model for each voice that we build, it would be a difficult exercise to find hand-annotated corpora of breaks appropriate to the required style. In fact, such corpora may not even exist for specific styles or languages we want to target. We thus require a method to train data-driven phrasing models using unsupervised learning methods.

We have proposed a novel grammar-based phrasing method to build style-specific phrasing models. Our method generates phrasing models that are accurate, style-specific, and completely data-driven. In this chapter, we shall look at the details of our proposed method and study how well it performs to baseline methods both objectively and subjectively. Later on, in Chapter 6 we shall look at how we can improve upon this method in order to build phrasing models when dealing with voices in resource-poor languages.

2.1 Previous Phrasing Techniques

To build and evaluate phrase prediction models, we need text data that is annotated with breaks. The TOBI standard (Silverman et al., 1992) is one of the most widely adopted systems for annotating prosody. The TOBI recommendation specifies annotating each word boundary with a numeric level between 0 and 4. Level 0 is used in cases of clear phonetic marks of clitic groups. Level 1 denotes a non-break boundary. Level 2 marks apparent disjuncture between words. Level 3 marks a intermediate phrase break (within an utterance), and Level 4 is a major break, such as at the end of an utterance. Several corpora have been annotated with these break indices and have been used in phrasing research. Two of these corpora, used commonly for English are the Boston University Radio News Corpus (Ostendorf et al., 1995), and the MARSEC (Roach et al., 1993) corpus.

While the TOBI recommendation specifies four levels of breaks, many phrasing models have not included the Level 2. Festival, for example simplifies this TOBI scheme to include just the levels 1, 3 and 4. Model predictions primarily decide whether a word boundary is a non-break (Level 1) or a break (Level 3). Festival detects end of utterances and places a Level 4 break at utterance boundaries.

Given an annotated corpus, there are several ways of building phrasing models. In their literature review, Read and Cox (2007) describe two main types of approaches to phrasing: (i) Deterministic, or Rule-Based, and (ii) Data-driven. Deterministic models are easy to build, require no training data. Rule based models, such as (Bachenko et al., 1990) can be successful, but are unreliable, often difficult to write, and difficult to adapt to new domains or languages. Data driven techniques use large amounts of training data and machine learning algorithms for classification.

The simplest deterministic model is a punctuation-based model: Where there is a punctuation, insert a break. Taylor and Black (1998) demonstrated that this model is extremely precise, but has only about 50% recall. This model is extremely easy to build, and is applicable to many languages. This is why Festvox implements this model as the default phrasing model for voices in new languages. The trouble with this model is that under some conditions, punctuation is unreliable or non-existent. Informal writing such as email or tweets often lacks punctuation. Tasks such as speech-to-speech translation often require us to synthesize text without punctuation. Nonetheless, this model provides a usable baseline especially for low resource languages, and we will use this model as a baseline later on in this thesis.

Data-driven phrasing has emerged as one of the most successful methods. Various machine learning methods have been proposed for phrasing, among them: decision trees (Wang and Hirschberg, 1992; Koehn et al., 2000), transformational rule learning (Fordyce and Ostendorf, 1998), hidden Markov models (Taylor and Black, 1998; Schmid and Atterer, 2004), memory based learning (Marsi et al., 2003), Bayesian networks (Maragoudakis et al., 2003), maximum entropy models (Liu et al., 2008), and neural networks (Ying and Shi, 2001).

Prosodic phrases tend to be balanced in length. This means, when doing a left-to-right prediction of phrase breaks, the decision of our classifier should depend on where the previous breaks were. Models that use only local features (such as part-of-speech tags) rely on other mechanisms to optimize the global phrasing over the utterance. For example, Taylor and Black (1998) use a 7-gram language model of break sequences, and Schmid and Atterer (2004) use the distance-from-previous-break feature to parameterize their HMM model. An alternative strategy to optimize utterance-level phrasing is to move a level up, from parts of speech, to syntax and phrase structure.

The relationship between prosody and syntax is not well understood. Prosodic breaks do not always correspond to syntactic breaks. Bachenko et al. (1986); Fach (1999) suggest that traditional syntactic phrase structure is not directly applicable for prosodic phrasing. Koehn et al. (2000); Read and Cox (2007) have shown that with the help of a high-accuracy parser, adding syntactic information can have significant improvements in phrasing. In more recent work, Saychum et al. (2011) have shown that a categorial grammar based model can be successful. Prahallad et al. (2010) have shown that prosodic and syntactic breaks can be different for utterances. They showed that prosodic breaks are speaker specific. A given unambiguous sentence will typically only have one linguistic phrase structure. At the same time it can have multiple prosodic phrase structures. It thus seems that syntax, in the sense of prosodic phrase structure, is a more promising path than linguistic phrase structure, at least from the point of view of phrasing. Liu et al. (2008) have used this notion, but modeled it differently from a parsing approach.

Speaker specific, or style specific models have not been the focus of most of the described work above. Phrasing models are typically trained on large data sets, such as MARSEC (Roach et al., 1993). Little has been done in the direction of style-specific modeling. The obvious problem to building a speaker-specific model is data scarcity. Many synthetic voices we build are from as little as half an hour of data. Sometimes the cost of gathering more data is high, and other times, it is just not possible (e.g., voice of a deceased person). It is important to build phrasing models that adapt to new conditions with limited amounts of data available. Bell et al. (2006) assume that the variability in phrasing is due to the underlying distribution of phrase lengths, and they have proposed an enhanced version of the (Schmid and Atterer, 2004) model that adapts the phrase length distribution to new domains. This method still solely relies on part-of-speech tags and does not take into account deep syntactic information. Obin et al. (2011) have proposed a method that uses segmental HMM and Dempster-Shafer fusion to incorporate linguistic and metric constraints in phrasing. This work uses a large coverage syntactic and morphological lexicon for French, along with a factored parser. They have proposed to test this model for different speaker styles.

2.2 Phrasing Evaluation Methods

Before we look at our methods of building phrasing models, it is important to cover the subject of evaluation: How can we tell whether the models that we shall build are better? Speech synthesis is a direct consumer facing technology: people use it as a medium to access text content via speech. One synthetic voice is truly better than another voice only if the users of our systems think it is so. Subjective evaluation is therefore very critical to speech synthesis. The same is true for phrasing. Our contention is that phrasing will help improve overall prosody of synthetic speech, and the primary method to evaluate that would be subjective listening tests. Evaluation is also integral to building and optimizing our models, but it is impractical to have subjective listening tests as part of model building: they are expensive and take time. Therefore, we need some objective metrics to evaluate phrasing: ideally, a metric that is correlated with human judgments.

In this section, we shall look at some objective metrics and subjective strategies to evaluate phrasing models.

2.2.1 Subjective Evaluation Methods

Subjective evaluation in speech synthesis typically consists of producing speech examples for a few prompts, and asking human participants to listen to them and grade them. Two of the most common listening tests are the Mean Opinion Score (MOS) test, and the A/B test.

In a mean opinion score test, a participant listens to one utterance at a time. They are asked to rate each utterance on some defined criteria, on a scale like 1 to 10. If participants grade two different systems, their aggregate opinion could help decide which system is better.

An A/B test is a direct, pairwise comparison of two systems. Participants listen to pairs of utterances at a time. The pair contains the same utterance produced by different systems. Participants are asked to pick the system that they think better fits a certain criterion.

In this work, we have used A/B tests as the main subjective evaluation method. We build two synthesizers, one that has our phrasing model, and the other that uses a baseline phrasing model. We use a set of prompts and synthesize them with the two voices. An A/B test is then performed, and we ask people which version they prefer. For each utterance, subjects submit their vote for a particular model, or they can choose a third option (no preference). By summing the votes over all utterances and participants, we can decide if our model is better by comparing the votes it received with those of the baseline.

In our A/B tests, we have deliberately asked for "preference" of listeners. Arguably, we could directly ask people which phrasing strategy is better. However, the direct question is tricky for two reasons: (i) Participants would need to be speech experts to understand what phrasing is, and how to judge it, and (ii) The overall idea is that phrasing could help the overall prosody, and by asking people to pay attention to just phrasing, we could ignore the evaluation of prosodic artifacts introduced, and their impact on naturalness and intelligibility.

Running subjective tests is time consuming and expensive, and using crowd-sourcing platforms like Amazon Mechanical Turk can help mitigate both these issues. We have therefore run most of our listening tests on such platforms. In order to simplify and standardize the workflow of setting up such tests, and to ensure that participants on the web can properly access the tests, an opensource software called TestVox has been created. This is an important contribution of this thesis, and is described in more details in Appendix A.

2.2.2 Objective Evaluation Methods

Phrasing is a classification problem: predicting each word boundary as being a break or not. Thus, classification accuracy is an objective metric we can use to evaluate our models. However, because phrasing is also a sequence labeling problem, we can also evaluate it by looking at the similarity or dissimilarity of the phrase length histograms.

One measure of accuracy is the number of correct predictions. Indeed, Taylor and Black (1998) measured the percentage of breaks correct, and non-breaks correct, and the total word boundaries correctly predicted. However, breaks and non-breaks typically have a very skewed distribution in a corpus. The number of non-break word boundaries is very high. Thus, a model that predicts all boundaries as non-breaks is likely to get a very high total accuracy. Looking at the individual accuracies on breaks and non-breaks is not ideal, because comparing two models that way is fairly difficult.

This problem can be resolved by calculating the accuracy in terms of precision and recall scores, and then combining them into a F-measure (van Rijsbergen, 1979). Precision tells how many of the predicted breaks are correct. Recall tells how many of the actual breaks were predicted. The combined harmonic mean value is thus a good indicator of overall quality of a model. If P is the precision, R is the recall, and F_1 is the F1-measure, then:

$$P = \frac{\text{Number of breaks correct}}{\text{Number of breaks predicted}},$$

$$R = \frac{\text{Number of breaks correct}}{\text{Number of breaks in test set}},$$

and

$$F_1 = \frac{2PR}{P+R}$$

Two models can be compared on their F-measure and the model with a higher score would be better. This is the standard comparison metric used in recent literature. In case a model does not predict any word boundary as a break, then the precision and recall are both zero and F_1 is undefined. However, we shall assign a F-measure of zero in that case.

We present another objective method to evaluate phrasing models: look at the distribution of phrase lengths. Phrase length is the number of words between two consecutive phrase breaks. After predicting breaks on a test set, we can find the overall phrasing profile: the histogram of phrase lengths. We can also build another histogram of phrase lengths on the actual phrases present in the test data. We can then measure a distance between the two histograms. We will use two of the commonly used metrics to compare histograms: the L2 distance (Euclidean) and the Earth Mover's distance (EMD) (Rubner et al., 1998).

If we represent two histograms as single dimensional vectors, then the L2 distance between them is simply the Euclidean distance between those vectors. If a and b are two histogram vectors, then:

$$D_{L2}(a,b) = \sqrt{\sum_i (a_i - b_i)^2}.$$

The EMD distance (Rubner et al., 1998) between two distributions is proportional to the minimum *work* required to change one distribution into another. We normalize our histograms so that they represent probability distributions of the phrase lengths. For distributions in one dimension, such as our histograms, Cohen (1999) has proved that the EMD between them is the area between the graphs of the cumulative distributions.

A fully accurate model would get both the L2 and EMD to be zero, and when comparing two models, a model with the lower distance from truth would be deemed better.

Given that phrasing model affects other prosody models, we can also objectively compare two models by synthesizing with them on held-out speech data and measure the Mel-Cepstral Distortion (MCD) of the synthesis, as defined in (Mashimo et al., 2001). The MCD metric is often used to judge the quality of synthesized speech. Calculation of MCD requires a time-alignment of the two speech samples, which can be done using Dynamic Time Warping. Similar to Prahallad et al. (2010), we can use the MCD to evaluate how the new phrasing model compares to another model.

Thus, in summary, when comparing two phrasing models objectively, we can look at the F-measure, the L2 distance, the EMD distance and the MCD distance. Models with the higher F-measure and lower distances would be deemed to be better.

2.3 Grammar Based Phrasing Method

Having looked the different phrasing strategies used in the past, and the evaluation methods to compare different phrasing models, let us now look at one of the main contributions of this thesis. In this section, we present our data-driven phrasing model that captures style-specific phrasing.

We use the Festival speech synthesis system (Black and Taylor, 1997) for our research. Our proposed method lies within Festival's framework. Festival's current phrasing method, described by Taylor and Black (1998), uses two models: (i) A POS sequence model, and (ii) A phrase break sequence model. If b_i is the probability of a break at juncture i, C_i is the context of observed features at juncture i, and B_i represents the context of previous break sequences at juncture i, then we want to estimate $P(b_i|C_i,B_i)$.

Using the Bayes theorem, we have:

$$P(b_i|C_i,B_i) = \frac{P(C_i,B_i|b_i) \cdot P(b_i)}{P(C_i,B_i)}.$$

Assuming that the events C_i and B_i are independent, we get:

$$P(b_i|C_i,B_i) = \frac{P(C_i|b_i) \cdot P(B_i|b_i) \cdot P(b_i)}{P(C_i) \cdot P(B_i)}.$$

We can rewrite that as:

$$P(b_i|C_i,B_i) = \frac{1}{P(b_i)} \cdot \frac{P(C_i|b_i) \cdot P(b_i)}{P(C_i)} \cdot \frac{P(B_i|b_i) \cdot P(b_i)}{P(B_i)}.$$

Applying Bayes theorem to the two right terms gives:

$$P(b_i|C_i,B_i) = \frac{P(b_i|C_i) \cdot P(b_i|B_i)}{P(b_i)}.$$

Festival uses the Taylor and Black (1998) model that models the term P(b_i|C_i) as a part-of-speech quad-gram model, and the term P(b_i|B_i) as a 7-gram language model. The term P(b_i) is the unigram probability of breaks as found in the training data. To obtain the best sequence of break/non-break tags over the entire utterance, Festival uses a Viterbi search to combine these models. Figure 2.3 shows this arrangement. The grammar based model that we are proposing in this thesis is a replacement for the POS sequence model in Festival. Instead of looking at just the POS sequence in a context, we use a combination of several features in a CART tree. Our CART tree is trained to predict breaks from features, and thus provides us with the probability $P(b_i|C_i)$.

At each word boundary, our grammar-based model predicts the probability of a break, considering the context of features. Like the POS sequence model in Festival, we deal with just two levels of breaks: a break (TOBI Level 3), or a non-break (TOBI level 1). While phrasing models typically do not explicitly dealt with TOBI level 2 breaks because of their difficulty to label and analyze, one could apply our method to incorporate this extra category into the classification. We use the two TOBI levels and run Festival's Viterbi search together with a phrase-break sequence model to find the best path of break/non-break tags that maximizes the probability of breaks in the entire utterance.

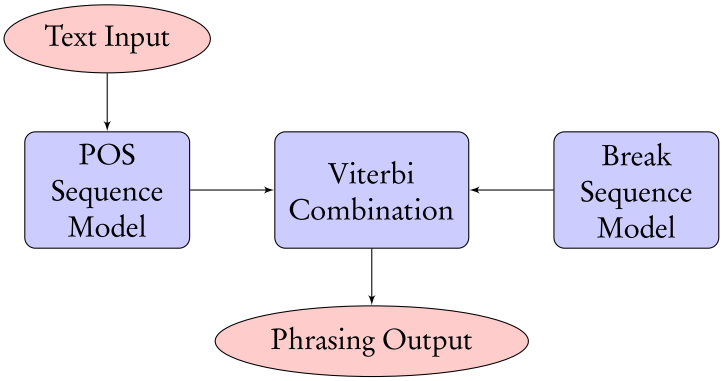


Figure 2.3: Festival's phrasing strategy. Our proposed model replaces the POS sequence model.

Our method is outlined in Figure 2.4. To train our models, we first automatically annotate text data with breaks. We run part-of-speech tagging over the text. We introduce bracketing, or phrase level chunking over our training utterances. We then train a Stochastic Context-Free Grammar (SCFG) (Pereira and Schabes, 1992) that can parse unseen utterances into similar prosodic chunks and introduce bracketing on them. We then extract features over this prosodic phrase structure and use our training data to build CART trees for phrase prediction. At test time, we replace words with their parts of speech, and run the trained SCFG to generate a prosodic phrase structure over them. We then extract the syntactic features and plug them into the trained CART tree to predict breaks. In the following subsections, we shall look at these different components in detail.

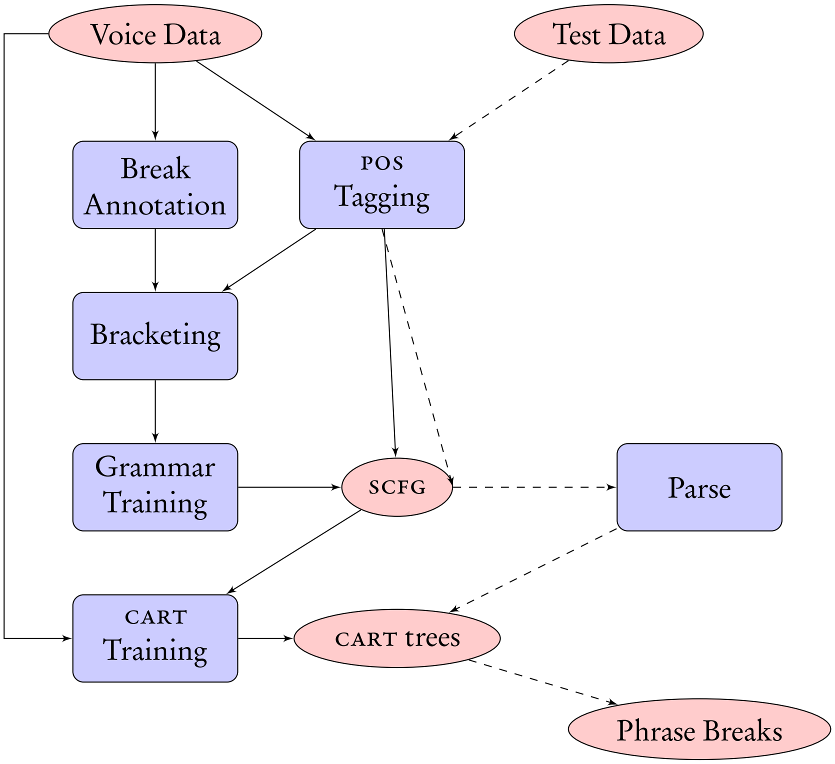


Figure 2.4: Overview of our approach. Solid lines show flow of training data. Dashed lines show flow at test time.

2.3.1 Automatically Annotating Training Data

Like all classifiers, we need some annotated data to train our models. Unlike previous phrasing work however, we choose not to use a hand-annotated corpus. Instead, we use acoustic data to derive phrase breaks in a corpus. When building synthetic voices, we typically have a few minutes or more of speech data with transcripts available. We believe that this speech not only captures the speaker's spectral identity, but also their phrasing traits. By training models on just this speech data, our hypothesis is that we will be able to learn the phrasing style that is specific to the speaker and the domain at hand. The other advantage of using only this speech data for training purposes is that this data is available for any language we are building voices for, and hence we can easily build models for any language. (Prahallad et al., 2010) have suggested a method to automatically extract phrase breaks from speech data. We use a similar method here.

We start with the speech database we have. We use the HMM alignment tool by Prahallad et al. (2006) and force-align this speech data to the transcripts. The alignment tool uses an optional silence HMM state at every word boundary and hence during Viterbi decoding. If there is a pause in the acoustic data, it will be marked in the labeled transcription automatically. We identify the pause regions in the signal. If there is no pause between two words, we mark the boundary as a non-break. If there is a pause that is longer than a certain threshold, we mark the boundary as a break. The choice of the pause threshold is important. A value that is too high would lead us to ignore genuine pauses from being labeled properly. A value that is too low would mark events such as glottal stops as being real pauses. Our empirical analysis on several corpora found that the threshold of 80ms is an appropriate threshold and we used this value in our experiments.

Running the forced alignment over the entire speech corpus results in annotated data that can be used to train a classifier.

We looked at how well this automatic labeling method performs, compared to hand-labeling of data. The Boston University Radio News Corpus (Ostendorf et al., 1995) has speech recordings as well as hand annotated breaks. Using a portion of this corpus as reference annotations, we measured the performance of our automatic break annotation. We found that the forced alignment method is 100% precise, but has a low recall of 55.88%, leading to an overall F-measure of 0.717. The breaks that the algorithm misses are typically shorter than 80ms in duration (our threshold), or have noise in the silence segments that gets aligned with adjacent phones. More sophisticated forced-alignment techniques could help improve the overall F-measure of automatic break labeling, but are beyond the scope of this thesis. The labels that our method yields are very precise, and allow us to train phrasing models for specific styles where hand annotated data is not available.

2.3.2 Part of Speech Tagging

Building a classifier over lexical items is not a good idea, because the vocabulary size would be too large, and dealing with out-of-vocabulary words would be an issue. We need a way to collapse the vocabulary down to a small finite set. To do that, we use Part of Speech (POS) tags. For English, we use the built in POS tagger in Festival, which uses the tagset from the Penn Tree Bank (Marcus et al., 1994). Festival also has a very small set of tags called *guessed* POS tags, which can also be used in our model.

2.3.3 Training the Prosodic Grammar

We train our prosodic grammar using the (Pereira and Schabes, 1992) algorithm implemented in the SCFG tools within Festival. The requirement for this algorithm is a bracketed corpus. We first convert our text data into a bracketed form, and then use the training tool to generate the SCFG.

We start with a text sentence in our training database. After determining word boundaries that are breaks, we use brackets to divide the text into prosodic chunks. For example, the sentence *there are five people in this room* may be chunked as *((there are) (five people) (in this room)).* The bracketing step is thus a fairly simple step. The SCFG training runs over the bracketed training data to train a prosodic grammar.

The prosodic grammar thus learned can be used to predict prosodic parses of new utterances. The grammar uses POS tags as the terminal symbols, and generates non-terminals automatically. We typically limit the number of non-terminals to 5, 10 or 20, depending on the amount of training data we have. The grammar induction algorithm is an iterative algorithm and we usually allow up to 20 iterations during training.

Figure 2.5 shows an example parse that we are trying to induce. This example is made up for illustration purposes. Notice that while in spirit we are similar to learning a linguistic phrase structure, the end result is very different from something like a grammar trained on, say the Penn Tree Bank (Marcus et al., 1994). Some of the constituents in this parse (NT0 in our example) fail linguistic constituency tests, and the non-terminal symbols have no particular linguistic meaning. The syntax simply tries to capture prosodic constituents.

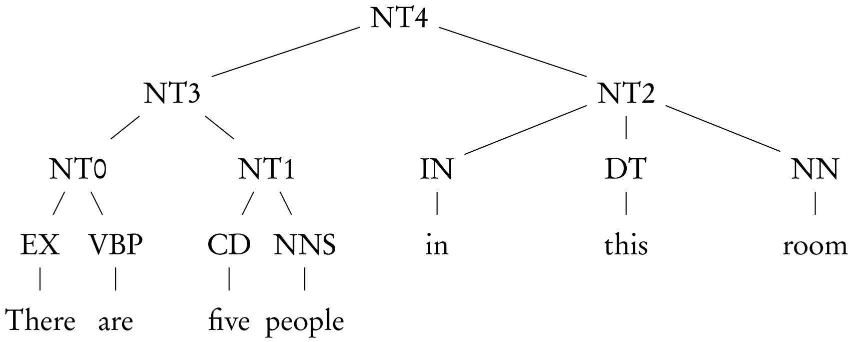


Figure 2.5: Example of an SCFG parse for prosodic phrases

2.3.4 Learning the CART prediction trees

Once we have built a grammar from our style specific training data, we parse our entire training set with the that grammar. We then dump features for each word in the corpus, including features about their positions with respect to the prosodic phrase structure predicted by the grammar.

With the word-level features and the truth value of break/no-break, we train a CART classification tree using wagon. We optimize the trees for the F1 accuracy. We typically use a 80/10/10 split of the available data between training, development and testing. The exact proportion of this split is not critical, as long as we ensure there is sufficient data in each split.

Table 2.1 lists the features that we included in our CART training. To take context into account, we use these features for the current word, two previous words, and two next words. After building trees on all the datasets, we looked at the top features in the their respective trees. The features “has-punc”, “end-brackets”, “delta-brackets” and “gpos” seem to be the ones carrying a lot of information about the breaks.

Table 2.1: CART model features

Name	Description
pos	Part of Speech
gpos	Guessed Part of Speech
has-punc	Is word followed by punctuation
lpunc	Current token is punctuation, but not next
token-in-quote	Does a single quote appear in this or previous token? (Disambiguate end-quote from possessive)
dist-to-eos	No. of words before sentence end
Grammar:	
end-brackets	Count end-brackets in prosodic parse
start-brackets	Count open-brackets in prosodic parse
delta-brackets	(scfg-end-brackets) − (scfg-start-brackets)
abs-delta-brackets	abs(scfg-delta-brackets)

2.3.5 Phrase Prediction for New Sentences

To use our models for new utterances, we first tag all words in the utterance with the appropriate POS tags. We use the trained SCFG grammar to induce bracketing over the tag sequence. We then extract all the relevant lexical and syntactic features out for the utterance and use the trained CART tree to predict breaks at each boundary.

2.4 Style Specific Phrasing

The main objective of our phrasing approach was to build speaker/style specific phrasing models. We looked at five different data sets belonging to different styles of speech. We built phrasing models on them as described, and compared them to the default model in Festival. In this section, we shall first look at the different styles, and analyze the corpora for stylistic differences. Then we shall look at the objective and subjective results obtained by our models.

2.4.1 Corpora and Styles

We looked at five different corpora that have speech in different styles.

The ARCTIC corpus consists of a half set, called A-set, of the ARCTIC prompt set (Kominek and Black, 2004) recorded by speaker AUP (an Indian English speaker). The style of this corpus is “short sentences”. The corpus has 593 prompts with an average of 9 words per prompt and the audio size is about 30 minutes.

We took the *Europarl* parallel corpus (Koehn, 2005) between English and Portuguese. This data contains proceedings of the European Parliament. We selected prompts from the English side of the corpus. These prompts were also recorded by speaker AUP. The style of this corpus is “parliament proceedings”. The corpus has 595 prompts with an average of 14 words per prompt and the audio size is about 50 minutes.

The *F2B* corpus is from the Boston University Radio News Corpus (Ostendorf et al., 1995). The style of this corpus is “radio broadcast”. The corpus has 464 prompts with an average of 19 words per prompt and the audio size is about 55 minutes.

The *Obama* corpus consists of public talks by the US President Barack Obama. Audio and transcripts of two of his public addresses were used to build this voice: (i) 2009 Presidential candidate speech “A more perfect Union”, (Mar 2008, Philadelphia) and (ii) Address at the Military Academy (Dec 2009). The style of this corpus is “public address”. The corpus has 465 prompts with an average of 18 words per prompt and the audio size is about 61 minutes.

The *Emma* corpus (Prahallad and Black, 2010) is taken from an Audiobook (Emma, by Jane Austen) in the Librivox database. The book was recorded by a female volunteer. The style of this corpus is “audio book”. The corpus has 9936 prompts with an average of 15 words per prompt and the audio size is about 1040 minutes (over 17 hours).

2.4.2 Analysis of Phrase Breaks

The different styles of speech appear to vary with respect to the global distribution of breaks versus non-breaks. We measured the percentage of word boundaries where a break was found in the original recordings for each dataset. Note that we excluded the breaks at the end of the utterances. Table 2.2 shows that while the ARCTIC, Europarl and F2B datasets have a similar proportion of breaks in them, the Obama and Emma data have more breaks. The table also shows how many breaks were globally predicted by festival's default phrasing model on each dataset. The numbers show that the default model is inserting more breaks than expected. To see why this may be the case, we looked at the MARSEC (Roach et al., 1993) data from which the default phrasing model is trained. That data has 14.15% of the word boundaries marked with breaks.

Table 2.2: Percentage of breaks in corpus

Dataset	Total Words	Actual Breaks	Default Predictions
ARCTIC	5313	6.25 %	8.96 %
Europarl	8066	6.48 %	11.28 %
F2B	9214	6.37 %	14.30 %
Obama	8402	9.21 %	14.50 %
Emma	158209	8.27 %	16.19 %

It turns out that the styles we are looking at don't differ just in the proportion of breaks, but also the distribution of durations of the breaks. We looked at the histograms of break durations on the datasets and observed that breaks in ARCTIC-A and Europarl are of similar lengths, whereas Emma and F2B have longer breaks on average. The Obama corpus has many long breaks, and also a long tail of breaks that go well over half a second in duration. The duration distributions are not truly Gaussian, and we shall look at them in more detail in Chapter 4. However, analyzing the means and variances of the duration values can help us see that the styles differ quite a bit in the duration. Table 2.3 summarizes the parameters of these distributions.

Table 2.3: Duration in seconds of pauses in recorded speech

Dataset	Mean	Stdev
ARCTIC	0.115	0.059
Europarl	0.111	0.067
F2B	0.273	0.099
Obama	0.391	0.311
Emma	0.180	0.162

2.4.3 Experimental Results

We built phrasing models on all five styles of speech. We used both the POS and the g-POS (Festival Guessed Part of Speech for English) tags as features in the CART. We ran preliminary experiments with the F2B corpus and found that using Festival's reduced POS set instead of the entire Penn-Tree Bank set provided optimal results. In fact, these experiments also helped us decide using 10 non-terminals in the grammar we trained.

We present objective results in terms of both the F-measure, and the MCD score of synthesis. To measure the accuracy, we held out 10% of our training data, which was automatically annotated for breaks. We compared our predicted breaks to the automatic annotations and measure the F-1 accuracy. We also synthesized the utterances and measured MCD with respect to the original speaker's recordings. All our data sets, with the exception of the Emma set are small in size. To obtain meaningful results, we performed a 10-fold cross validation on them. The Emma data set was large, and building models was computationally expensive, and we did not cross-validate on that data.

Table 2.4 shows the comparison in terms of the average F-measure between the standard Festival phrasing and our approach. We observe a significant improvement on all but the *Obama* data set. This shows that our model is able to capture the different styles appropriately, across not only the different styles but also different data size scenarios. We have noted, in Table 2.3 that the break distribution of the *Obama* corpus, especially in terms of the break durations was much different than others. The corpus also has applause in the speech data, which leads to difficulty in analyzing it. Having a very high variance in the pause lengths, and especially a long tail of many long pauses is likely to cause errors in the HMM forced-alignment step which forms the very basis of our phrasing method. This could explain why the new method did not outperform Festival for the *Obama* corpus. Note however, that the difference between the baseline and our method in this case is not statistically significant.

Table 2.4: Phrasing Accuracy (F1 score) on Different Styles. Bold values significant for $p < 0.005$

DataSet	Festival Default	Our Method	Improvement
ARCTIC	80.11	85.12	**5.01**
Europarl	70.42	77.67	**7.25**
F2B	66.17	73.67	**7.50**
Obama	66.41	63.80	-2.61
Emma	69.98	82.94	**12.96**

Table 2.5 shows the comparison in terms of the average MCD of the synthesis using Festival default phrasing, compared to synthesis using our approach. We see a consistent improvement in the synthesis across all the styles. Note that the absolute numbers in this table are slightly on the higher side, compared to scores on similar data in the literature. This is because this is the MCD of full synthesis, not just re-synthesis with natural durations, and we are using Dynamic Time Warping that shifts the MCD numbers on the higher side.

We conducted listening tests to compare the proposed model to Festival's default model. We selected 25 random sentences from the heldout test set and synthesized them using either of the phrasing models. We made sure that the phrasing decisions for these utterances, as predicted by the two models are not identical. We then did an A/B test over Mechanical Turk, and asked participants to listen to two versions of every utterance at a time, and mark which one they prefer. We filtered responses that we could predict as being workers trying to make easy money by submitting random answers and collected statistics to see how many times participants preferred our proposed model. Participants were also allowed

Table 2.5: Comparison of MCD of synthesis on Different Styles. Bold values significant for $p < 0.0005$

DataSet	Festival Default	Our Method	Improvement
ARCTIC	7.47	7.18	**0.29**
Europarl	7.12	6.67	**0.45**
F2B	6.20	5.95	**0.25**
Obama	10.25	10.08	**0.17**
Emma	6.98	6.60	**0.38**

to say that they could not prefer one model over another. Table 2.6 shows the votes that the two models received in our listening task. We see that people can perceive the differences between these phrasing models, and that they prefer the proposed, grammar based phrasing model over the Festival baseline in all but the Obama voices.

Table 2.6: Subjective Preference (% responses) of Synthesis on Different Styles.

DataSet	Festival Default	Our Method	No Preference
ARCTIC	33.3	48.8	17.9
Europarl	41.5	51.5	7.0
F2B	37.7	48.6	13.7
Obama	48.7	47.3	4.0
Emma	40.0	53.0	7.0

2.5 Chapter Summary

In this chapter, we described one of the most important contributions of this thesis: the grammar based phrasing method. We first described the phrasing problem as a classification task. We discussed that phrasing patterns are specific to speakers and styles. We looked at the details of how our data-driven grammar based method works, and our results show that the method is better than the traditional model used by Festival across different styles with varying amounts of training data available.

Our grammar based phrasing method has a data-driven approach. It uses automatically annotated, acoustically derived phrase breaks for its training. We learn and use a stochastic context-free grammar to introduce syntax in our utterances: syntax in the form of prosodic phrase structure, as opposed to linguistic phrase structure. This model replaces the traditional part-of-speech sequence model in Festival. Details of this model have been published in (Parlikar and Black, 2011). Code for this model is now integrated with the Festival and Festvox suite of tools with documentation and is available for further experimentation. The presented grammar based method is flexible, and not specific to the phrasing problem: Anumanchipalli (2013) has successfully applied it to the problem of predicting accent groups from syllables.

Chapter 3

Minimum Error-Rate Phrasing

> "Vary the pace..." is one of the foundations of all good acting.
>
> Ellen Terry

Speaking styles have a lot of variety. While some styles are clearly distinct, such as an audiobook versus a newsreader, some styles are only slightly different. For example, a newsreader could be speaking at a fast or a slow pace. In these variations within a style, the phrasing is also affected. Slow newsreaders pause more often while fast news readers omit several pauses. If corpora with such variations in speaking rate were available to us, we could train style specific grammar based models and use them appropriately. However, we usually do not have corpora that are explicitly recorded for variations of style, and we need flexibility in our phrasing models to change the phrasing rate to suit speaking rate, even if explicit training data was not available. More over, we desire to have not just 'fast' and 'slow' phrasing, but a continuous number that denotes exactly how much phrasing is happening.

The Grammar based phrasing model presented in the previous chapter is just one of the components of the phrasing process. The other parts consist of a break sequence model, and Viterbi combination of these two models, as shown in Figure 3.1. In this chapter, we propose a new architecture for phrasing, that involves a log-linear combination of multiple phrasing models, and a process called minimum error-rate training. We will see that this new architecture not only provides us with flexibility in modeling phrasing, but also improves accuracy of prediction on top of improvements that the Grammar-based method provides.

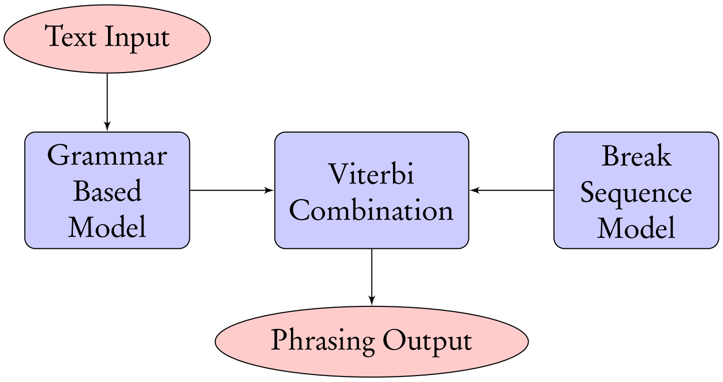


Figure 3.1: Festival's phrasing architecture we are proposing to change

3.1 New Phrasing Architecture

Phrase breaks are only occasional events in speech. In about an hour of speech, such as the F2B data, there are about 600 breaks. When training a phrasing classifier under such conditions, we are likely to encounter the issues of having little training data. We showed that our proposed Grammar-Based model can deal with this data sparsity situation well. However, adding complexity to our model in the future would be difficult because estimating all parameters with the little training data would be hard. Under limited training data conditions, machine learning methods typically consider building multiple classifiers (some of them could be potentially weak), and then combining them together to make final decisions.

One weak point of the default Festival phrasing architecture is that it allows for two fixed models: a model that predicts the probability of a break given context of features, and a model that looks at the sequence of predicted breaks. These probabilities are the $P(b_i|C_i)$ and $P(b_i|B_i)$ respectively.

We are introducing a new architecture for phrasing in Festival, that allows us to go beyond the two fixed models. Our system allows adding multiple phrasing model to the mix of making final phrasing decisions. It also supports arbitrary features that can be used to our advantage, such as to vary phrasing rate as we will see later in this chapter.

Our proposal is to use a log-linear model for phrasing. A key advantage of log-linear models is that they allow a very rich set of features to be used in a model. Let us assume that we are given a text sequence $\mathbf{t}$, and we want to produce a break sequence $\mathbf{b}$. Among all possible break sequences, we will choose the sequence with the highest probability:

$$\mathbf{b}^* = \arg\max_b P(\mathbf{b}|\mathbf{t}).$$

We directly model the posterior probability $P(\mathbf{b}|\mathbf{t})$ using a log-linear model. In this framework, we have a set of M feature functions, $h_m(\mathbf{b}, \mathbf{t})$. For each feature function, we have a weight w_m. The direct phrasing probability is then given by:

$$P(\mathbf{b}|\mathbf{t}) = \frac{\exp\left(\sum_{m=1}^{M} w_m h_m(\mathbf{b}, \mathbf{t})\right)}{\sum_{b'} \exp\left(\sum_{m=1}^{M} w_m h_m(\mathbf{b}', \mathbf{t})\right)}.$$

The modeling problem here is to define suitable feature functions that capture the relevant properties of the phrasing task. The grammar based phrasing model, as well as the break sequence model that the old Festival architecture uses can very well become two of these feature functions. In addition, we can add the POS sequence model from Taylor and Black (1998). We can also define other arbitrary features. Note that the features don't have to be probability distributions themselves. A schematic depiction of the new architecture is shown in Figure 3.2.

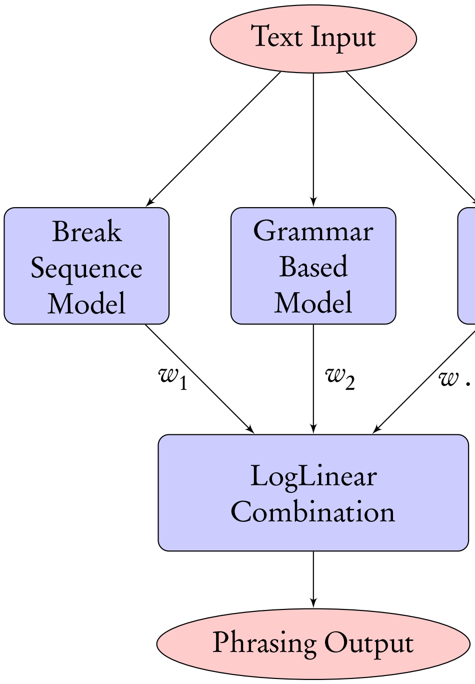


Figure 3.2: New proposed phrasing architecture in Festival

3.2 Minimum Error-rate Training

We need to train our loglinear model in order to use the defined features appropriately. The training problem is to find out the suitable weights w_1^M. Typically, these weights are determined by maximizing the likelihood of the model over development data. We could train our model for maximum likelihood. However, eventually, the model will be objectively evaluated on a metric such as the F-1 score (van Rijsbergen, 1979), because an improvement in this score is usually perceived as an improvement also in subjective listening. In order to make a higher perceptual impact, we aim to optimize our loglinear phrasing model directly to the F-1 score. The idea of using Minimum Error-Rate Training (MERT) in training this phrasing model is inspired from its use by Och (2003) and others in the field of Statistical Machine Translation.

We define a held out development corpus D_1^N, of size N, with text sequences T_1^N that has reference break annotations R_1^N. Our goal is to obtain minimum error on this corpus, and a set of k different candidate break sequences, $S_n = \mathbf{b}_{n,1}, \ldots, \mathbf{b}_{n,k}$. That is, for each of the N sentences in the test set, we have k hypotheses of break sequences, and we want to pick the ones that minimize overall error on the test set. Given a set of weights w_1^M, the top-best break sequence $\mathbf{b}_n$ for sentence n is given by:

$$\mathbf{b}_n = \operatorname*{arg\,max}_{\mathbf{b} \in S_n} \left[\sum_{m=1}^{M} w_m h_m(\mathbf{b} | T_n) \right].$$

For each sentence in the development corpus D_1^N, we can pick the best break sequence given some weights, and then compute the F-measure over these break sequences. The error function E can then be set to negative value of the F-measure. If $E(D_1^N; w_1^M)$ represents the error on the test set given a set of weights, we have:

$$w_1^M * = \operatorname*{arg\,min}_{w_1^M} \left[E(D_1^N; w_1^M) \right].$$

The optimization criterion here is tricky. Because of the presence of an arg max operation within the error function, we can not compute the gradient of the error, and hence an optimization method such as gradient descent can not be used here. The error surface is not smooth, and has many local minima.

We use the Basin-Hopping algorithm by Wales and Doye (1997) to optimize the error function at hand. This global minimization method has been shown to be extremely efficient for a wide variety of problems, and is especially useful when the error function has many minima separated by large barriers. In particular, we use the implementation of this algorithm within the Python SciPy toolkit.

We use a development corpus, use a randomly initialized weight sequence and produce an n-best list of break sequences. We then run the minimum error rate training over these n-best sequences and learn new weights. We then use the new weights and re-generate an n-best list of break sequences over the development corpus, and run minimum error rate training again. We repeat these iterative process until the final error does not improve across an iteration. After each iteration, we also normalize the weight vector to be of a unit norm.

3.3 Phrasing Improvements

We used four feature functions $h_m(\mathbf{b}|\mathbf{t})$ in our method. Two of these are the same as the models in the baseline phrasing method: (i) The context model $P(b_i|C_i)$ that looks at the lexical and syntactic context, and (ii) $P(b_i|B_i)$ that looks at the language model probability of the break sequence. In addition, we use another context model, $P(b_i|C_i)$, defined in Taylor and Black (1998) that looks at the part-of-speech tag context at a word boundary and uses a quadgram Language Model to predict the probability of the word boundary being a break. Finally, we use a break-count feature, that counts the total number of breaks in the predicted break sequence.

We evaluated our method on two synthetic voices trained using the CLUSTERGEN (Black, 2006) statistical parametric synthesis method: (i) Voice built from the F2B corpus within the Boston University Radio News Corpus (Ostendorf et al., 1995), and (ii) Voice built from two hours of recordings of Jane Austen's books, for Blizzard Challenge 2013 task EH2.

Our baseline phrasing models were built to be grammar based style-specific phrasing models in each case, as described in the previous chapter. We trained the proposed model with minimum error rate method on a held out corpus, and used an unseen test dataset in the same domain to compare the baseline method to the proposed approach. Table 3.1 shows the comparison. We see that on both the styles, we observe an improvement in phrasing accuracy.

Table 3.1: Objective Evaluation (F-score) of the Proposed MERT Method. Improvements are significant at $p < 0.05$

Voice	Baseline	MERT
F2B	54.35	58.06
Audiobook	52.87	57.58

3.4 Phrasing Rate Knob

One requirement of a phrasing model is that it should be flexible to adapt to the speaking rate of a synthesizer. A slow synthesizer should probably mark more word boundaries as breaks, and a faster synthesizer can do away with a few breaks. If the user of a speech synthesis engine demands that 30%, or 60% of the word boundaries should be breaks, then our phrasing model should be able to meet this requirement. However, this is a tricky constraint. If our training data had splits corresponding to slower and faster speaking styles, we could train individual classifiers and use the appropriate one at synthesis time. But such data is seldom available, and collecting data to train such specific models is difficult. We describe how we use the log-linear framework and MERT mechanism to provide a knob, a continuous number, to vary the number of phrase breaks produced.

One of the features that we used in the log-linear model was simply the number of phrase breaks in a given break sequence. This feature allows us to define a knob to change the number of phrase breaks our model produces.

Intuitively, the break-count feature tries to make sure that the number of breaks produced by our model is reasonably close to the number of breaks in the reference sequences in our development data. Even if we optimize towards the F-measure of break prediction, which itself balances precision and recall of phrasing, having this additional feature means that the weight learned for this feature will try to produce an optimal number of phrase breaks. If we keep the weights for other features to be the same, and change the weight of the break-count feature, then the search process at synthesis time picks utterances with more or fewer breaks than the optimal. For example, if we subtract a number from the weight of the break count feature, and maximize the log-linear combination, we would produce more breaks. We can vary the value of this weight and measure the number of word boundaries in a development corpus that were breaks. The weight of the break-count feature is thus the knob we can use to tweak the amount of phrasing. Figure 3.3 shows this curve for the two voices we have. The x-axis shows the value of the knob (i.e., the weight of the break-count feature) and the y-axis shows what percentage of word boundaries in a corpus were predicted as being breaks.

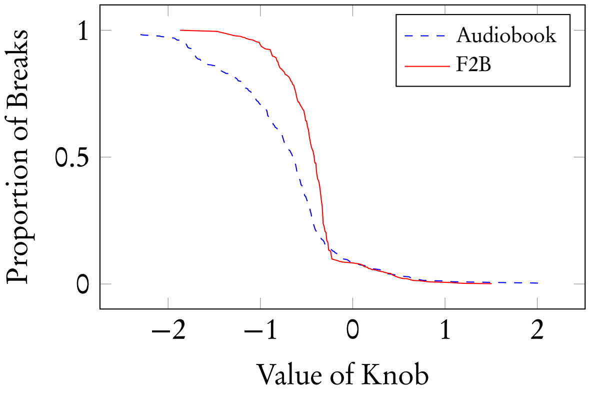


Figure 3.3: Proportion of phrase breaks generated by varying the log-linear weight of the break-count feature (the knob)

In order to customize the phrasing rate at demand, we need to parameterize the "knob", so that given a particular value of expected proportion of breaks, we can set an appropriate weight for the break-count feature during synthesis. This problem boils down to deriving an equation for the inverse of the function represented in Figure 3.3. Given a particular phrasing proportion x, we want to find out the value k that our knob should be set to.

To learn the parametric equation of the knob, we use a development corpus and varying values of the weight of the break-count feature to generate data points depicted in Figure 3.3. We then fit the data automatically to a variety of sigmoidal, trigonometric and simple functions and choose the function that best fits the data we have, as measured by the root-mean-squared error of the fit. We used open-source fitting code, pyeq2 by (Phillips) in this work.

Our empirical analysis shows that for various corpora and phrasing model combinations, the phrasing knob curve can be approximated, well within a RMSE tolerance of about 0.15, using a Tangent equation with offset:

$$k = A \cdot \tan\left(\pi \frac{x - C}{W}\right) + O$$

where A (Amplitude), C (center), W (width) and O (offset) are the parameters we learn automatically. For the F2B voice, we obtained

$$k_{f2b} = -0.165 \cdot \tan\left(\pi \frac{x - 0.5073}{1.0772}\right) - 0.4670$$

and for the Audiobook voice, we obtained

$$k_{audiobook} = -0.2682 \cdot \tan\left(\pi \frac{x - 0.5048}{1.072}\right) - 0.8084$$

Figure 3.4 shows the real curve of the phrasing proportion generated by our loglinear model, along with the tangent approximation curve, for the F2B corpus. Figure 3.5 shows the same graph for the audiobook data. To read these graphs, start by picking a number on the x axis. This is the knob value that our tangent estimate will predict, for a particular value of desired proportion of breaks. Then follow the vertical line corresponding to the x value, and look at the difference between the blue and red lines. This shows the error that we make in predicting the appropriate proportion of breaks. Visual analysis of the curves of F2B data shows that apart from the region of strong inflection in the actual curve, towards the bottom, the blue and red lines have very little error between them, thus supporting the claim that the tangent function provides a good estimate overall to predict the knob value. The error in the case of the audiobook is overall larger than that of F2B, for the central region.

By tweaking the knob to change the phrasing rate, we deviate from the reference break sequences that we originally used to train our MERT model. This means, by changing the knob, we obtain fewer or more breaks, but at the cost of the F-measure. Of course, since the goal was to insert more or fewer breaks, the penalty in F-measure is not very relevant anymore, but we looked at what the drop in the F-measure looks like. Figure 3.6 shows how the F-measure changes when we set the expected break proportion to different values. We observe that the F-measure is highest when the knob is set to its original value, as learned from the MERT training.

3.5 Perception of Phrasing Rate

The MERT phrasing model, with the knob feature allows us to vary the phrasing rate. For each utterance, we can set an expected break proportion,

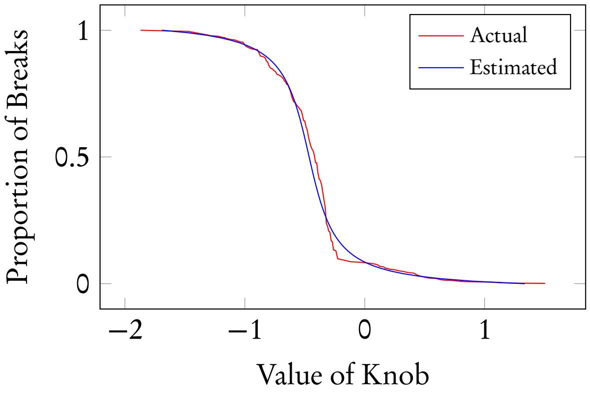


Figure 3.4: Comparing actual knob values to values estimated using the Tangent function, on the F2B corpus.

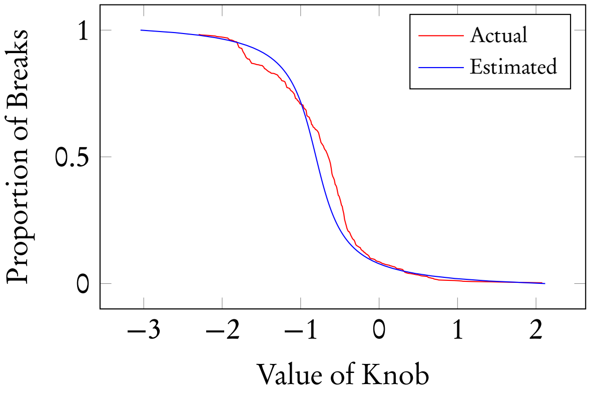


Figure 3.5: Comparing actual knob values to values estimated using the Tangent function, on the audiobook corpus.

i.e., how many word boundaries should be breaks, and we saw how the tangent approximation function of the knob could be used to generate the desired proportion of breaks. We investigated how these variations in phrasing rate affect perception of speech.

Understanding the perception of phrasing rate is important to synthesizing appropriate variations of phrasing. The Mel scale studies showed that the human ear is not equally sensitive to differences in pitches at the

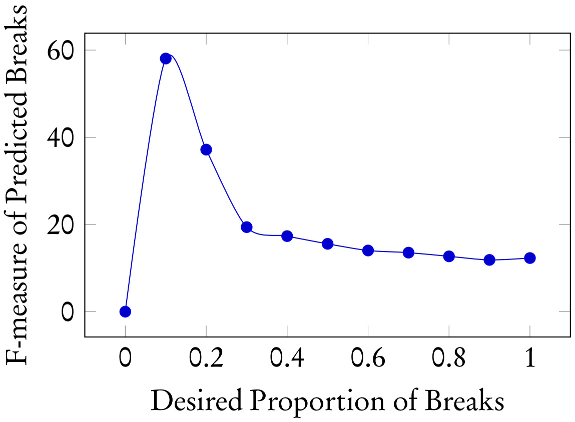


Figure 3.6: F-measure versus Desired Proportion of Phrase Breaks on the F2B corpus

lower end of the audible spectrum and at the higher end. In a similar spirit, we wanted to evaluate whether people can distinguish between different phrasing rates, and how sensitive they were to the differences.

We used the F2B voice described above to perform listening tests. We configured the voice to use eleven knobs, each to produce an expected phrasing rate of 0%, 10%, 20%, . . . , 100%. Phrasing rate of 0% means that no word boundary is a break, and a 100% phrasing rate means that all word boundaries are breaks. We synthesized utterances from a held out test set using these eleven knob values. We chose three utterances, longer than 10 words each, as our data for running perception study. For each utterance, we have 11 variations, thus we have 33 waveform files of synthetic speech in total.

We compared the different phrasing rates pairwise. We compared the 0% phrasing rate individually with 10%, 20%, 30%, 40%, and 50%. Similarly, we compared the 10% phrasing rate to 20%, 30%, 40%, 50% and 60%. Thus each phrasing rate was compared with up to five phrasing rates on the higher and lower side. Overall, we had 40 pairs of phrasing rates to compare. For each comparison, (say 10% with 40%), we used the three utterances and played them to listening test participants, and asked them whether they think the utterances are same, or different. For each utterance in each pair, we asked ten participants to submit their responses. This provides a total of $40 * 3 * 10 = 1200$ responses. The listening test was conducted on mechanical Turk, and we had included gold standard questions to weed out spammers from submitting their responses.

What we wanted to know was: how much should a phrasing rate change, if people are to notice a difference? For each pair that we compared in the listening test, we can calculate what fraction of the responses thought the pair was same, versus different. Thus, for each pair, we can get a number between 0 and 1, where 0 means the pair is clearly the same, and 1 means the pair is clearly different. We can then draw a 11x11 grid, comparing the similarity or difference between each pair of phrasing rates. We can fill each cell with a color (a heat map) that shows how similar or different they are. Figure 3.7 shows this data.

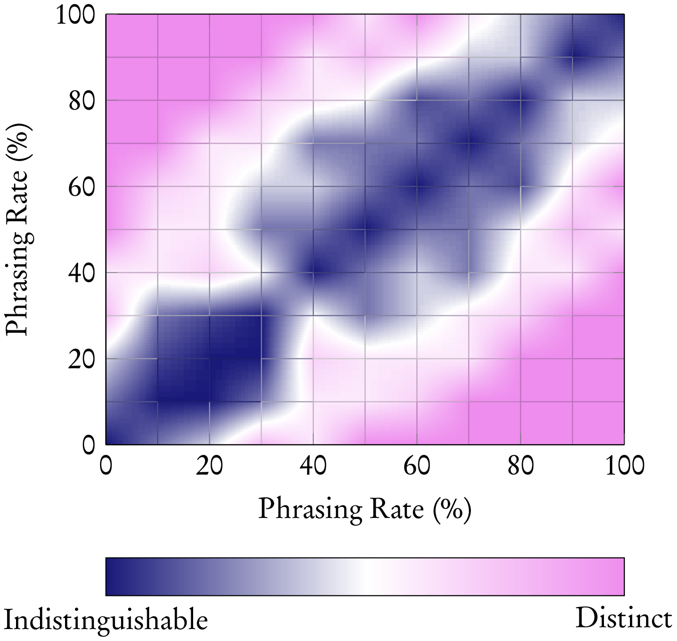


Figure 3.7: Heatmap showing perceptual measure of distinction between two phrasing rates

To read and interpret Figure 3.7, start with any value on the *x* axis.

Now follow along the y axis for that value. For the phrasing rate at the chosen x value, any phrasing rate y that is blue in the figure will be be perceptually similar. Pink cells will be perceptually distinct. Notice the diagonal cells are dark blue, because synthesis using a certain phrasing rate will clearly be perceptually indistinguishable from itself. There are two important observations we can make from this figure: (i) Starting from any phrasing rate, if we want to change the phrasing rate to be something different, we need to manipulate our knob quite strongly. For example, starting from a phrasing rate of 20%, perceivable difference in phrasing can only be made by doubling the phrasing rate! (ii) As a corollary, if we tweak the knob only a little bit, and change the phrasing rate by 10-20%, then the resulting phrasing will be perceptually indistinguishable. This can be of advantage if extra breaks need to be inserted for domain-specific reasons. One example where this is useful is to achieve lip-sync during video dubbing. See Section 5.2.5 for a discussion on this.

A visual inspection of the heatmap shows that there are three regions, bordered at a phrasing rate of 40%, and 80%. We also note, from Table 2.2 that normal speech has breaks in the vicinity of 10%. If we had to make the speech slower, we would have to up the phrasing rate to 40%. If we would like to make it even slower, we would have to then up the phrasing rate to 80%. If we had to make the phrasing knob a fixed scale, rather than a continuous value, our hypothesis was that we can use the values corresponding to the default, 40%, and then 80%. We tried to evaluate whether people can classify these different phrasing rates as normal, slow, and very slow. We did this with the help of a listening test. With 10 utterances and 10 subjects evaluating each utterance on mechanical Turk, we played the different phrasing rates independently, and randomly. We asked people to classify each audio clip as normal, slow and very slow. Participants were given examples before the test began so they could understand the space of differences. Table 3.2 shows somewhat mixed results of this perception study. We observe that when presented with the default, "natural" phrasing rate, people agree a lot on their view of the speech being normal. When we up the phrasing rate to 40%, people are somewhat certain that the speech is not 'very slow', but they are divided on their opinion of whether it is 'normal', or 'slow'. Moving the phrasing rate up to 80% seems to leave people into a state of no consensus — the opinion is almost equally divided between 'normal', 'slow' and 'very slow'. Our hypothesis of the 10-40-80 phrasing rate groups for a 'fixed knob' is only mildly supported by this evidence. While people can tell these phrasing rates apart, they do not seem to agree on what label to assign.

Table 3.2: Subjective Listening Test: Can people classify phrasing rates into 'normal', 'slow' and 'very slow' categories?

Phrasing Rate	% Votes Normal	% Votes Slow	% Votes Very Slow
Default	77	21	2
40%	50	33	17
80%	28	38	34

3.6 Chapter Summary

In this chapter, we described our method of defining the phrasing problem under a log-linear framework and training the framework with a minimum error rate target, rather than maximum likelihood. We showed that combining features/models related to phrasing using this MERT strategy produces a significant improvement in phrasing accuracy, as measured by the F-1 metric.

We described a break-count feature integral to our MERT model that allows us to define a parametric "knob" to vary the quantity of generated phrase breaks. Once we learn our MERT weights, we can keep all weights to their learned value and vary the weight of the break-count feature to provide this knob. Our empirical evidence shows that the knob can be reasonably approximated with a Tangent function with offset. The combination of using a MERT model and this break-count feature allows a user to specify how many breaks they want, and our model produces the breaks appropriately.

Our model now can vary the phrasing rate at demand. We map a particular phrasing proportion into a knob value. However, users of speech synthesis do not define phrasing rate in terms of the proportion of word boundaries that are breaks. The control we would like them to have would be more quantized: low, medium, high, etc. We studied the perceptual impact of phrasing rate (proportion of breaks) and saw that people are quite insensitive to small changes in phrasing rate. This means that large changes in phrasing rate are necessary if we want to make a difference in perception. But more practically, it means that we can get away with little changes for specific reasons.

The MERT framework that we proposed for phrasing took inspiration from work in Machine Translation. However, this connection actually runs deeper. Text to speech is often used as the final step in speech to speech translation, and we are required to synthesize automatically translated output. We shall get back to this model in Section 5.2.3 with the objective to improve the intelligibility of synthesized translations.

This proposed new architecture of phrasing in Festival has been published in (Parlikar and Black, 2013). Code for this model has been integrated in the Festival and Festvox suite of tools with documentation. The underlying framework is extensible from phrasing to other models within the CLUSTERGEN synthesis paradigm, and future work could look into applying this to other prosody models.

Chapter 4

Predicting Break Duration

The right word may be effective, but no word was ever as effective as a rightly timed pause.

Mark Twain

Pauses in speech are not all the same. The obvious difference is that pauses at end of sentences are longer than those within a sentence. However, the differences are more subtle than that. In the previous chapters, we saw how phrase break prediction can be made style specific. However, there is more to style than just the positions of breaks. Goldman-Eisler (1961) has observed that lengths of individual pauses in speech are distributed differently for different individuals, as well as the type of situation in which speech is uttered. Oliveira (2002) has discussed about work that demonstrates that people tend to pause more often and remain in silence for much longer when speaking in more complex scenarios. Duration of pauses can affect perception. For example, Dhillon (2008) shows that pause duration is a reliable means of discriminating between lexically ambiguous words. From a synthesis point of view, it is thus important to not just model where we insert prosodic breaks, but also predict the duration of these pauses.

Although phrase break prediction has been widely explored in synthesis, generating these breaks with the appropriate duration has not received much attention. Generally, all duration models treat the pause separately. Some segmental duration modeling techniques, such as Campbell (1992) does not predict pauses at all. Barbosa and Bailly (1997) divides an utterance into rhythmic groups and predicts the duration of each group. It computes the segmental duration of the group and then optionally inserts a pause of the remainder length. Following the Klatt (1982) model, the Festival speech synthesis system (Black and Taylor, 1997) as well as the Mary TTS system (Schröder and Trouvain, 2003) assign a fixed duration to breaks, based on the predicted TOBI level of the break. One of the reasons why pause specific duration models have not been thoroughly explored is that style corpora with appropriate annotations are not easy to construct. The BURNC corpus (Ostendorf et al., 1995) is one such corpus, but it hasn't been widely used to build pause duration models.

In this chapter, we present a data-driven approach to modeling duration of phrase breaks. Similar to methods in previous chapters, we use forced-alignments between speech and transcription to detect where phrase breaks are in natural speech. For each break, we find out its duration and extract features over text that could be used to learn a regression predictor for the duration. Here we present our work on six data sets, which vary both in size and in speech style.

4.1 Styles and Corpora

The stylistic corpora we used for duration modeling are the same as those we used in Chapter 2. We used an additional audiobook corpus. Here is a recapitulation of the corpora we used.

The *Europarl* corpus consists of prompts from the English side of the Europarl (Koehn, 2005) parallel corpus between English and Portuguese. This data contains proceedings of the European Parliament. The speech was recorded by an Indian English speaker (AUP) in the style of "parliament proceedings". The *ARCTIC* corpus consists of the ARCTIC prompt set (Kominek and Black, 2004) recorded by speaker SLT (female, American speaker). The style of this speech is “short sentences”. The *F2B* corpus is from the Boston University Radio News Corpus (Ostendorf et al., 1995), in the style of “radio broadcast”. The *Obama* corpus consists of public talks by the US President, Barack Obama. Audio and transcripts of two of his public addresses were used to build this voice: (i) Presidential Candidate speech (Mar 2008, Philadelphia) and (ii) Address at the Military Academy (Dec 2009). The *TATS* corpus is taken from an audio-book (The Adventures of Tom Sawyer, by Mark Twain) in the Librivox database. The book was recorded by a male professional volunteer. This is in the “audio-book” style. Finally, the *Emma* corpus (Prahallad and Black, 2010) is taken from an audio-book (Emma, by Jane Austen) in the Librivox database. The book was recorded by a female volunteer. The style of this corpus is also, broadly, “audio-book” but is different from the *TATS* corpus.

We extracted the pause duration from natural speech in our corpora. To do that, we force-aligned the speech and transcriptions using an EHMM tool by Prahallad et al. (2006) that allows for short silences to be inserted during the alignment. We used these alignments to find out the length of these inserted silences. We ignored all inserted pauses that were less than 80msec in length.

Table 4.1 shows the break duration profile of these corpora. Note that we do not include breaks at the end-of-utterance in our analysis here. This is because, for some databases, the end-of-utterance pause timing may no longer be in the database due to external pre-splitting, and/or recording being done as isolated sentences. We observe from this table, however, that the average break duration varies quite a bit across the styles.

4.2 Analysis of Phrase Break Duration

As argued by Campione and Véronis (2002), it is better to use log-transformed duration, rather than values in the time domain, to analyze or model pauses. This is because corpus studies have shown that the log-distribution is closer to being a normal distribution than the original distribution. We looked at the log distribution of the duration of breaks extracted from our corpora.

Figure 4.1 plots the kernel density estimates of the log-distribution

Table 4.1: Break Profile of our corpora. Counts here do not include breaks at ends of utterances.

Corpus	Speech Size (minutes)	Num Breaks Per Minute	Average Break Length (msec)
Europarl	49	7.2	141
ARCTIC	56	3.8	130
F2B	55	10.6	273
Obama	61	11.8	414
TATS	406	7.6	249
Emma	1040	8.5	243

of breaks that we extracted. We observe that the distribution of breaks between different corpora is quite different. Also observe that even in the log domain, our break distribution is far from normality. We can perhaps consider that the break duration values come from a Gaussian mixture. Using the Benaglia et al. (2009) approach, we looked at each corpus and analyzed how many Gaussian distributions the mixture for that corpus consists of. The analysis shows that ARCTIC, Obama and TATS corpora consist of a mixture of two Gaussians, the Europarl and F2B are mixtures of three Gaussians, whereas the Emma distribution is best expressed as a mixture of four Gaussians. The analysis by Campione and Véronis (2002) described trimodal distribution of pauses, and categorized them as brief (<200ms), medium (200-1000ms) and long (>1000 ms). In our analysis, we have completely excluded the end-of-utterance breaks, which are typically long pauses. Even if we assume that those breaks are normally distributed, the entire set of pauses in our case does not always seem to be trimodal.

4.3 CART Duration Modeling

Our analysis showed that there is a lot of variation in the duration of phrase breaks in natural speech, that we should attempt to capture during speech synthesis. To predict duration of each phrase break, we built a decision tree regression model. Each of the leaf nodes of the tree makes the assumption that the breaks within its context are normally distributed

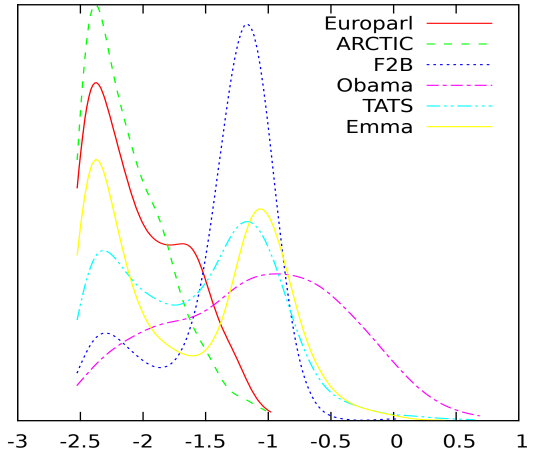


Figure 4.1: Kernel Density Plot of log-duration of breaks

(i.e., we store the mean and variance of the breaks). While this seems in violation of our analysis that the breaks are in fact not normally distributed, our contention is that the questions asked by the decision tree would sufficiently narrow the context of the breaks so that all breaks in that context fall into a distribution that can be approximated with a Gaussian estimator.

We started with all the breaks extracted from each of our corpora. We dumped a set of features corresponding to the breaks and used *wagon* to build regression trees. Like in our other experiments, we used an 80-10-10 split of data between training, development and testing. We used the 'stepwise' option that the *wagon* program provides, and this allowed us to select the most informative feature at every level by evaluating it on the development set when building the tree.

At synthesis time, Festival first predicts the positions of all phrase breaks in an utterance and then builds the duration of segments from left to right. In this particular work, we used the standard phrasing model in Festival, the Taylor and Black (1998) model to perform the phrase break decisions. When predicting the duration of a segment that corresponds to a phrase break, we use the custom tree trained to predict the break duration.

The set of features we used in our modeling is as follows: (i) name of the two segments before and after the break, (ii) part of speech of the two words before and after the break, (iii) punctuation character, if any, before and after the break, (iv) presence of a quotation mark before or after the break, (v) the number of content words in the previous phrase, (vi) the number of stressed syllables in the previous phrase. From building models and analyzing the output of the build process, we made the following observations. The type of punctuation that occurs before the break is a great predictor for break duration. The names of the adjacent segments and the parts of speech of adjacent words are also good indicators of break duration. The number of words or syllables in the previous phrase was not consistently a useful feature for predicting the duration of breaks. Analysis by Zvonik (2004) suggests that having more than 10 syllables in the preceding phrase strongly correlates with a long break. In the styles and corpora that we used, however, we did not notice such a strong relationship between the length of the previous phrase and the duration of the break. Note however, that the prediction of where breaks should be inserted in the first place does depend on the length of the previous phrase.

4.4 Evaluation of Break Durations

In the earlier chapters, where we looked at the prediction of where phrase breaks should be inserted, we measured the accuracy objectively by using metrics such as the F-measure. Duration prediction is a regression problem, not a classification problem. Hence, we need to use some other metrics to objectively compare how well our models are doing.

One of the commonly used metrics in duration modeling (at segment level, or otherwise) is the root-mean-squared-error (RMSE) of prediction. If we use the durations of breaks in our held-out data, $d_1, d_2, \ldots d_n$, as being the ground truth, and if our model predicts $p_1, p_2, \ldots, p_n$ as the actual duration values for those breaks respectively, then the RMSE is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(d_i - p_i)^2}{n}}.$$

The other commonly used metric for duration modeling is correlation between the truth and predictions. The sample correlation coefficient can be used to estimate the Pearson correlation r between the true durations of breaks $d_1, d_2, \ldots, d_n$ and the predicted respective values $p_1, p_2, \ldots, p_n$, by using the equation:

$$r = \frac{n \sum d_i p_i - \sum d_i \sum p_i}{\sqrt{n \sum d_i^2 - (\sum d_i)^2} \sqrt{n \sum p_i^2 - (\sum p_i)^2}}.$$

The goal of modeling break duration, at least objectively, is to achieve a low RMSE value, and a high correlation with truth. However, these objective metrics are not very strongly correlated with perceptual judgments, and we would also like to compare our models based on whether people think they are different, and/or better.

4.5 Experiments and Results

We have six corpora at hand. We extracted breaks and related features from each of the corpora. We then partitioned this data into 10 sets, with the intent of doing a 10-fold cross validation. In every set, we held out 10% of the data for testing. Instead of taking every tenth item into our test set, we preserved the sequence of breaks in training and testing data.

For each cross-validation fold and for each corpus, we have four different models. The baseline model is what Festival uses by default: predict each sentence-internal break as being 150ms. We can make this model a bit smarter by building a "Mean" model: Instead of predicting 150ms, we can predict the mean value of breaks that we saw in the training data for that cross-validation fold. Third, we built a style-specific model as described before. Finally, we built a non-style-specific model, or a combined model: we combine the same cross-validation fold of all our corpora and train a combined CART model. The purpose of this combined model is to provide us with a reference performance of a model that is trained using method similar to our style-specific method, but still is not style-specific. The hope is that style specific models will be better than the generic, combined model.

For each fold in the cross validation, we estimated the RMSE and Correlation number of our prediction using the four models at hand. We then averaged out the results over all cross-validation folds and looked at the average result for each corpus.

Table 4.2 shows the RMSE error of the four models on each corpus. Table 4.3 similarly shows the correlation of prediction. The RMSE and correlation values are on the prediction of the duration in the log-domain.

Table 4.2: RMSE of Predicted Duration (log-seconds domain)

Corpus	Festival	Mean	Combined	Style Specific
Europarl	0.4099	0.3858	0.6167	0.4186
ARCTIC	0.3897	0.3313	0.6858	0.3422
F2B	0.6778	0.4433	0.4794	0.4360
Obama	1.0456	0.7199	0.8685	0.7491
TATS	0.6736	0.5934	0.6021	0.5934
Emma	0.7072	0.6563	0.5834	0.5697

Table 4.3: Correlation of Predicted Duration (log-seconds domain)

Corpus	Festival	Mean	Combined	Style Specific
Europarl	0.0000	0.0000	0.1939	0.0653
ARCTIC	0.0000	0.0000	0.2974	0.2174
F2B	0.0000	0.0000	0.1251	0.2770
Obama	0.0000	0.0000	0.0790	0.0868
TATS	0.0000	0.0000	0.2276	0.1885
Emma	0.0000	0.0000	0.4634	0.5096

Looking at the RMSE, we see that the style specific model performs better than the Festival model on all but the Europarl style. It is also better than the combined model. However, quite often, the mean model seems to get a lower RMSE. This is a bit surprising because it is a naive model, and moreover, the underlying distribution of breaks is not even normal. One possible explanation here is that since there is little training data on most of our corpora (one or two breaks in every utterance), our models are over-fitting to the training data across all cross-validation folds, leading to weaker final model. If we look at the Emma corpus (our largest corpus), we see the results as we would expect: the Festival baseline does the worst, the mean-prediction does slightly better, followed by the combined model, and the style-specific model has the least error.

While RMSE is an important dimension to consider for evaluating our models, achieving the right speaking style means we should get a good correlation measure too. The mean model predicts a fixed value, and hence is not correlated at all with the actual duration. Our CART models can have a good correlation. The combined model typically gets better correlation numbers than the style specific model, even though it typically has higher error. On the largest (Emma) corpus, however, we see that the style specific model has higher correlation than the combined model, as we would expect.

Objective results show that style specific duration is better than other models, on the Emma corpus. We ran subjective tests using this corpus to understand two aspects of duration modeling. First, we wanted to find out if people can even perceive differences in break duration for synthetic speech. Strictly speaking, we did not combine the pause duration into any other prosodic model (such as F0), and hence we wanted to investigate the impact of the duration alone, on perception. Secondly, if people can indeed perceive the difference between pause duration, we wanted to find out if they prefer to listen to synthesis that uses the style-specific duration model.

We ran two subjective comparisons. First, we compared the “Mean” model to the style specific model. We also compared the “Combined” model to the style specific model. We used a preference test for both these comparisons. We chose 25 utterances from our test data and synthesized them with the three models of phrase duration. In this case, we used Festival’s default phrasing model to predict the location of breaks. We then created two tasks on Amazon Mechanical Turk to compare the two pairs of models. In each task, we presented the 25 utterances in random order. The two versions of each utterance were presented in random order, and were not labeled. Workers were asked to select the version that they preferred. We allowed up to 10 workers to do our tasks, and thus for each pair of comparison we have up to 250 data points of comparison. We filtered out responses by listeners that our automatic heuristics flagged as being spammers. The model that received the most votes by listeners can be considered to be the better one.

Figure 4.2 shows that the the style specific model performed better than the Mean model. This suggests that people perceive and prefer variability of pause duration in speech. Figure 4.3 shows that people could not tell the style-specific and combined models apart. This could mean that people can not tell apart subtle differences in variable duration. It could be that people can notice the presence of variation in pause duration, but not distinguish the subtleties. This is not unlikely, because pauses only happen once or twice in an utterance, and someone paying attention to understanding the overall speech may not pick up on the small differences in duration of breaks. However, this needs further investigation.

Figure 4.2: Subjective Result: Listener Preference for the Style-Specific model versus the Mean model

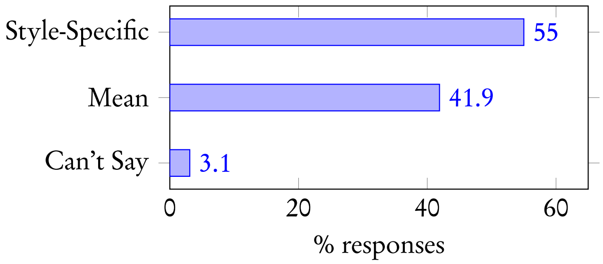
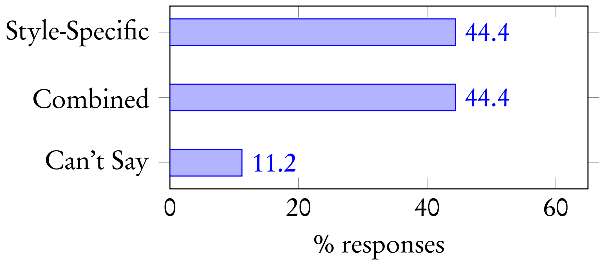


Figure 4.3: Subjective Result: Listener Preference for the Style-Specific model versus the Combined model



4.6 Chapter Summary

In this chapter, we looked at the duration aspects of stylistic phrasing. We looked at analysis of corpora which shows that there is a lot of variation in break duration between styles. Analysis also showed that not all styles have normally distributed break durations, even in the log domain, but that it could be modeled as a multimodal distributions.

We built decision tree models to predict phrase break duration for the styles with a data-driven approach. We compared these models to two naive models, that use fixed durations of breaks, and a non-style-specific model. On the Emma corpus, we found both objectively and subjectively, that the style-specific model was better than the fixed-duration models. The results on other corpora fixed-mean-prediction yields the lowest RMSE, but training a stylistic model gives variation in the break durations. The details of this work have been published in (Parlikar and Black, 2012b).

We realized, through efforts in the presented experiments, that predicting the duration of phrase breaks is a harder problem than predicting their location. This is primarily due to data scarcity. Unlike segmental duration, where many instances are available for each segment at training time, breaks occur less frequently in data. If speech corpus available to train a voice is about an hour of speech, the total number of breaks is likely to be far short of even 5000 data points, and building regression estimators over real numbers is challenging. However, in such situations, our results show that we can continue to use a fixed-value duration of breaks, but use a value that is learned from the corpus rather than an arbitrary constant such as 150ms. If larger data is available, such as in our case of the Emma corpus, then building the style-specific models is feasible and it would provide objective and subjective improvements over a fixed-value model.

In the work presented in this chapter, we only addressed phrase breaks that are inside an utterance. The next steps are to look at predicting breaks between utterances. Until recently, speech synthesis has usually focused on synthesizing one utterance at a time. However, paragraph synthesis is gaining popularity in domains such as audio-book synthesis. If synthesis happens at paragraph levels or higher, we have to start caring not just about breaks within an utterance, but also breaks between utterances in a paragraph, and breaks at the ends of paragraphs. Modeling the duration of these breaks is tricky, since databases from which we can train them are not readily available. Prahallad et al. (2007) have proposed a method with which large speech corpora could be aligned to their text, to automatically build a corpus for TTS voices that includes information at sentence and paragraph boundaries. One next step could be to construct such databases for available audio-books and model the duration of all phrase breaks occurring in speech. We would need to use cross-sentence features, and implement a support for them in the Festival Framework.

Chapter 5

Phrasing: Broader Scope and Impact

Technology feeds on itself. Technology makes more technology possible.

Alvin Toffler

Phrasing models are first of the prosodic models used by a speech synthesizer. The choice of particular phrase breaks can have an impact on other models of prosody, such as intonation and segmental duration. Further, phrase breaks also have an impact on the overall intelligibility of text. While the previous chapters looked into how phrase breaks can be modeled, this chapter focuses on the broader scope and impact of phrase breaks. We shall see how intonation modeling is affected by the choice of phrase breaks. We shall also look at speech synthesis in the context of speech to speech translation, and see how appropriate phrase breaks can make the synthesis of automatic translations more understandable.

5.1 Impact on Intonation

Phrasing is an important step during prosody generation in speech synthesis. It is also the first of the prosody models that gets invoked, and lays the foundation of prosodic decisions. Decisions that a phrasing model makes can have an impact on other models of prosody, such as intonation and duration. Improvements in the phrasing model should therefore be able to produce more natural speech synthesis.

We studied the interaction of our phrasing models with the Statistical Phrase Accent intonation model (Anumanchipalli et al., 2011). We built a voice using the data provided for Blizzard Challenge 2013, under the EH2 task. The training data consisted of ten hours of speech, and was taken from audio recordings of two books. The style of speech was thus, "audio book". We built a grammar-based phrasing model (Chapter 2) for this style and also built a minimum error rate phrasing model (Chapter 3) for the voice. We trained a Statistical Phrase Accent Model (SPAM) for the intonation. We used held out utterances to produce synthetic speech in two versions: versions: (i) Using the Festival default phrasing with the SPAM model, and (ii) Using the proposed phrasing model with the SPAM model. We looked at the intonation contours predicted by the SPAM model in the two cases, and compared them to the reference.

Our goal here was to study whether the SPAM intonation model predicted better pitch contours because of the introduction of the proposed phrasing model. Because we have reference pitch contours (from natural speech) over these utterances from the held out data, we can objectively evaluate the two predicted contours.

Two of the commonly used metrics in evaluating pitch prediction are the Root-mean-squared error (RMSE) and the correlation of prediction (Anumanchipalli et al., 2011). However, these metrics require time-alignment between the prediction and reference signals. Typically, pitch prediction is performed under "resynthesis" conditions, where the phrasing models and duration models are kept invariant, and different intonation models are then evaluated objectively. However, our case is a little different: we are changing the phrasing, thereby affecting both duration and intonation, and the predicted pitch contour is no longer time aligned with the reference from natural speech.

We used Dynamic Time Warping (DTW) to align the predicted and reference pitch contours. The DTW distance, in terms of the L2 norm, after the alignment is performed could be used to study whether phrasing affects the intonation model.

We first ran an experiment to study the validity of the DTW distance as an objective evaluation for improvements in pitch prediction. We built four voices, on the F2B corpus from BURNC. The intonation model in these four voices was trained on data with different sizes. Table 5.1 shows the DTW distance of the intonation model compared to reference natural speech on held out data. We know that an intonation model becomes worse as we reduce the amount of training data. This trend is reflected in the DTW distance. Therefore, the DTW distance seems to be a valid objective metric for evaluating different intonation contours that are not time-aligned.

Table 5.1: DTW distance of predicted F0 with models trained on different-sized subsets of F2B data. Bold values are significantly different from the row above. The trend supports the validity of DTW as an objective metric of F0 prediction.

Data Size	**DTW**
(F2B Corpus)	**Distance**
100%	26.1260
66%	26.1754
50%	**26.6518**
33%	**27.3685**

We now compared the intonation of the two audiobook voices we have: one with the default phrasing model, and one with the proposed model. Table 5.2 shows the DTW distance of the two intonation predictions with respect to natural speech. We see that introducing the proposed phrasing model significantly improves the SPAM intonation model.

Anumanchipalli (2013) has argued that an intonational model can be deemed better if it better captures the variation that exists in natural speech. One simple measurement they used is the mean and standard deviation of predicted pitch: a model that is closer numerically to the statistics in natural speech is better. We did similar comparison on the

Table 5.2: DTW Distance of F0 prediction with different phrasing models. Bold values significant for $p < 0.0001$.

Phrasing Model	DTW Distance
Festival Default	29.2358
Proposed Phrasing	**25.7285**

two voices we have. Table 5.3 shows that compared to natural speech, the intonation model that is based on the proposed phrasing model is slightly better than when using the baseline phrasing model. The difference here is not very big, but the trend is encouraging.

Table 5.3: Mean and Standard Deviation of F0 in synthesized and Natural Speech. The phrasing models proposed in this thesis help the intonation model get closer to natural speech.

Speech	Mean F0 (Hz)	Stdev F0
TTS: Festival Phrasing	174.926	28.418
TTS: Proposed Phrasing	178.607	30.582
Natural Speech	191.618	40.613

Overall, we found that the proposed phrasing techniques do impact the intonation in a positive way. We tried to evaluate the impact on a subjective level but found that designing a listening test that is valid is difficult. An informal A/B test measuring preference between the two voices (SPAM intonation model with the default and proposed phrasing methods) showed that the synthesis with proposed phrasing is better. However, this result is in line with the results found in the evaluation of style-specific phrasing models in Chapter 2. We don't know if the preference is because of a better phrasing model, or better intonation model. While the objective trends show that better phrasing also causes better intonation, a subjective listening test such as A/B, or even AB/X is not necessarily a valid test. This problem will hold for any study of interaction between different prosodic models, and a subjective strategy that can tease apart the improvements coming from individual models needs to be designed.

5.2 Impact in Speech Translation

One of the most interesting, yet challenging applications of speech synthesis is in speech-to-speech translation. In this task, we want to translate spoken utterances in one language, into spoken utterances in a foreign language. Ideally, we would want the translations to be exact in content, fluent in language, and delivered in the same style as the original speech. However, in a typical automatic spoken translation scenario, a person speaks into a microphone, and ill formed, difficult-to-understand translations come out of computer speakers. We decided to investigate into this problem and offer some solutions to make synthesized translations easier to understand.

Speech-to-speech translation (SST) is an inherently difficult task. It broadly consists of three modules. Speech data first goes through Automatic Speech Recognition (ASR) and gets converted into text. This text is then given to a Statistical Machine Translation (SMT) engine to convert it into text in foreign language. A text-to-text (TTS) system then reads out the foreign text. Although modern ASR and SMT systems are quite advanced, they are prone to errors. Further, errors in ASR can get amplified by errors in SMT. The result is that the output of the SMT engine often contains ill-formed sentences. A TTS system is supposed to read these ill-formed sentences, but standard synthesizers are trained on fluent text. They can not always handle the disfluent sentences correctly, and end up producing speech that amplifies the errors in text. In the end, we have speech that can be very difficult to understand.

Recent Blizzard Challenge results (see website) show that synthetic speech is still not as intelligible as natural speech. Further, Tomokiyo et al. (2006) have shown that synthesis of automatic translations is even less intelligible. Research is continuously improving the ASR, SMT, and TTS models individually, but comparatively little is being done to jointly improve the performance of this speech-to-speech translation pipeline.

Improving the overall SST pipeline involves deeper integration between its individual models. Improvements in translation accuracy have been obtained by tightly coupling ASR and SMT systems. Zhou et al. (2007); Bertoldi et al. (2007) and others have shown that using ASR word lattices or confusion networks as input to an SMT system can find a better translation rather than translating just the 1-best output of ASR. The link between ASR and TTS has also been looked at. Agüero et al. (2006); Sridhar et al. (2008) have looked at transferring prosodic information in the source speech onto the target side. This has shown to make the synthesis output more natural.

Our work here focuses on tighter integration between the SMT and TTS components. If an SMT engine produces output with fluency errors, a typical TTS system is not designed to handle it well. By letting the SMT and TTS modules talk to each other and communicate their strengths and weaknesses, they can jointly make better decisions about the final output. In addition, a TTS system, using the minimum error rate phrasing as proposed in this thesis can make phrasing decisions that can improve the intelligibility of machine translation.

5.2.1 Previous work: SMT–TTS Integration

There are two main issues with the boundary between SMT and TTS. First, that the output of SMT may not be grammatical or fluent. Secondly, the overall "dialect" of the language that the SMT generates bay be different than that the synthesizer was trained on, and hence even if the SMT output was fluent, TTS would have trouble synthesizing it correctly.

Ungrammatical and disfluent sentences are difficult to understand, even if presented in textual form. Tree (2001); Watanabe et al. (2008) suggest that when humans are reading such text, they explicitly mark disfluencies by adding silent or filled pauses at the appropriate places. These pauses alert the listener of an imminent phrase that may be difficult to understand. Taking this into account, Bonafonte et al. (2006) have used the probability of a word level language model as a feature in their TTS system in their phrasing model. Adell et al. (2012) have further shown that synthesizing filled pauses not only makes the utterances easier to understand, but also generates speech that is perceived to be more natural.

Paraphrasing input text is another technique that has been explored. Putois et al. (2010); Cahill et al. (2009) have suggested that for some tasks, the exact wording of an utterance is not crucial. A Paraphrasing tool can be used to generate several paraphrases of the actual input. The TTS system can choose any of the paraphrases, based on the unit-selection join cost. They used an SMT system that translates from one language into the same language, thereby creating an n-best list of paraphrases to choose from.

5.2.2 TTS-Friendly Translations

Sometimes, the output of an SMT system is grammatically and semantically fine, but is difficult to synthesize for our TTS models. For example, the generated utterance might have an unusual diphone at a word boundary. To mitigate this issue, we can use the n-best list from SMT instead of just the top best. This is similar in principle to (Cahill et al., 2009), but instead of paraphrasing given input text, we already have an SMT system at hand.

We start with the top-best translations we have to synthesize. For each utterance, we obtain the set of phonemes we would synthesize. We compare diphones in this set to the diphones in the training data of our voice. If an unseen diphone was found, we classify that utterance as having a bad join. We found that 17% of our translated sentences had a bad join in them.

For sentences that have a bad join, we seek to find an alternative translation from the n-best list. However, we need to ensure that by going down the n-best list we do not lose translation quality. We take the top-best sentence (the one with the bad join) as being a reference sentence. We compare each of the n-best items to the top-best and determine the METEOR score (Banerjee and Lavie, 2005). Our candidate hypotheses are the ones that have a METEOR score of 0.98 or above. We used METEOR instead of BLEU because of its reliability on an individual sentence level. We then go through the filtered n-best list and find the unit selection cost (Hunt and Black, 1996) of each hypothesis. We then pick the hypothesis with the lowest join cost.

To evaluate whether this n-best reranking gives us better intelligibility, we ran a transcription task. We picked 20 sentences that we knew had bad joins. Then we synthesized them, and also synthesized an alternative translation from the n-best list as obtained using the above method. We had five subjects listen to the utterances and we had them transcribe. We measured the error in the points of the bad join. We found that using the top-best translation had a word error of 28.9%, whereas using alternative hypothesis had a lower error of 24.7%.

5.2.3 Phrasing for Intelligible Translations

Appropriate phrasing can make synthesis easier to understand. We wanted to see if customized phrase prediction on SMT output can help improve its synthesis. We only had text output from the SMT system, and had to elicit phrase break information in order to perform this experiment. We had human annotators go through the SMT output and mark the word boundaries where they would insert breaks if they were reading the text out loud. We did not train a phrasing model, but used some of these annotated utterances, synthesized them and ran an oracle experiment to see if such customized phrasing can increase intelligibility.

The SMT output we chose for this experiment comes from a phrase based Chinese–English translation system. We chose this language pair because there is divergence in the word orders in these languages, leading to longer-distance word reordering errors in output. This system was trained on about 11 million parallel sentences and used the Gigaword corpus for language modeling. We used the moses decoder (Koehn et al., 2007) for translation. The test set we used was in the broadcast-news domain, and had a BLEU score (Papineni et al., 2002) of 14 points with one reference available for evaluation. We selected 80 translations that were between 10 and 20 words long. We manually removed 15 sentences that were very grammatical. Three people were asked to annotate phrase breaks in these sentences. All annotators are fluent English speakers and have a background in linguistics.

People don't always agree on phrase annotation, so we tried to determine the level of agreement between our annotators. We computed the kappa statistic (Cohen, 1960) between the annotations. On average, the annotators had a kappa value of 0.66. However, this substantial level of agreement can not be relied upon. Most word boundaries are non-breaks, and agreeing on them is easy. As argued by Stevenson and Gaizauskas (2000), this issue will affect other standard measures of inter-annotator agreement as well. It actually turns out that the agreement is not that high. From Table 5.4, we can see that there is a great difference in the number of phrase breaks inserted by each person. It is unclear how much of that can be attributed to personal phrasing preferences, and how much to the complexity introduced by the ungrammatical SMT output. Nonetheless, just like phrasing for fluent text, there clearly is no one-correct way of phrasing garbled SMT output.

Table 5.4: Annotator Agreement on Phrasing of SMT output

Annotator	Number of Annotations
A	99
B	59
C	88
A ∩ B	41
B ∩ C	31
C ∩ A	53
A ∩ B ∩ C	28

We then checked if customized phrasing can help improve the synthesis of SMT. We picked 20 annotated sentences (all from one annotator) and synthesized them with two phrasing versions: (i) The default Festival model, and (ii) Manually annotated phrases. The set of 20 sentences was chosen randomly, but care was taken to ensure that the phrasing generated by Festival isn't exactly the same as what our annotators marked (or we would not be comparing anything meaningful). We also made sure these sentences did not have hard words such as uncommon named entities. We asked five subjects to listen to these utterances and transcribe them. We post-processed these transcriptions to normalize case, correct typos and remove function words. We compared the transcriptions to actual text and calculated the word error rate. Table 5.5 shows the average transcription error of the two synthesis models. This result suggests that customized phrasing for SMT has potential to increase intelligibility and that we should pursue that goal.

We also looked at the SMT error of untranslated words. When SMT decoders encounter untranslated words, they can either leave them as they are, or delete them entirely. Untranslated words can cause problems

Table 5.5: Comparison of Annotated Phrasing to Standard Phrasing

Phrasing Model	WER
Festival	30.0 %
Customized	26.6 %

understanding content. If the source and target languages use the same orthography, the synthesizer can try to pronounce a foreign word and perhaps result in misleading speech. We had output of an Portuguese–English translation system that had several untranslated words. An example sentence: *It does raise problems "la" again.* With the assumption that untranslated words are likely to be a cause of trouble, we replaced them with filled pauses during synthesis. We synthesized the filled pause with an *um* sound at 20% lower pitch than the rest of the synthesis. No other changes were explicitly made to the overall prosody of the utterance. With a set of twenty sentences and four subjects, we ran a transcription test again. We compared the synthesis version with untranslated words left bare, versus the version with the filled pauses. In this case, it doesn't make sense to measure the overall word error rate, because transcribers are most likely to make error in the neighboring words of the untranslated word. Hence we measured error rate on the nearest two content words of the filled pause. We noticed that synthesizing untranslated words resulted in a 30.1% error. Using filled pauses lowers the error to 24.0%.

5.2.4 Automatic Phrase Breaks for SMT with MERT Phrasing

Given that appropriate phrase breaks can help the intelligibility of synthesized translations, we investigated whether we can automatically introduce such breaks with the help of a language model. We built a baseline voice on the F2B corpus. We trained a grammar based phrasing model. We built a MERT phrasing model in a manner similar to that described in Chapter 3, but we added an extra feature to the loglinear model: the trigram language model score at the word boundary (previous word, current word and next word). We trained an English language model on the Europarl data. For each word boundary, we queried the probability of the word sequence at the boundary.

We did the MERT training in three ways. First, we simply trained it on the F2B development corpus. In the second condition, we used a mix of the F2B development corpus and actual MT output that was hand-annotated with reference breaks (as described above). In the third condition, we only trained the MERT weights using the MT development data. We then evaluated the performance of the phrasing model on heldout portion of the labeled MT corpus. Table 5.6 shows the F-measure of the phrasing under these three conditions. We have also included results of the MERT training without the language model feature.

Table 5.6: F-measure of phrase break prediction, studying the impact of the language-model feature in MERT

3gr LM Feature?	Dev Corpus	F-1 (SMT)	F-1 (F2B)
No	F2B	10.64	58.06
Yes	F2B	12.63	58.06
Yes	F2B+SMT	29.03	56.90
Yes	SMT	41.67	35.62

We observe that the MERT phrasing model trained on the F2B corpus performs poorly when synthesizing machine translation output. If we add a language model to the MERT model, and still only train MERT weights on F2B data, we get a very small improvement on the SMT held out data. The intuition behind why the LM feature will help is that it will help detect bad word boundaries. However, the F2B development data is fluent English, and it doesn't have examples of "bad boundaries" that the MERT can learn from. By including SMT data to the mix of development corpus for training MERT weights, we get evidence of the bad boundaries and the language model feature suddenly becomes more discriminatory. This can also be seen by the weights that are assigned to the different models after optimization. Table 5.7 shows how the weights shift when tuning to the F2B corpus, versus the mix of F2B and SMT corpora. Notice the shift in the relative weight of the language model feature. Notice that when tuning to the mix of F2B and SMT corpora, we do not lose too much accuracy on the F2B test data. However, if we optimize the MERT only for SMT data, then in a way we make it specific to the SMT style, and the performance on the F2B data greatly goes down.

Table 5.7: Weights learned by MERT for different features when optimizing for different datasets.

Feature	F2B	F2B+SMT
Grammar Based Model	0.47756	0.48307
POS Sequence Model	0.12344	0.39139
3gr Language Model	0.03279	0.13300
Break Language Model	0.86617	0.75993
Break Count	-0.07327	-0.13519

We evaluated the impact of adding the language model subjectively, by running an A/B test between the base model (without a language model) to objectively the best model (mixed development data). We synthesized 25 utterances and asked 10 participants on mechanical Turk to listen to each. For each utterance, they heard the synthesis using two models. We asked them to choose the utterance they thought was more understandable. The two audio versions were presented in random order every time. Figure 5.1 shows that people prefer the proposed model.

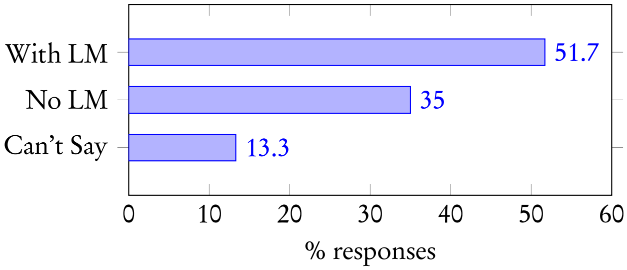


Figure 5.1: SMT Phrasing Subjective Test: Which model produces more understandable translations?

The inclusion of SMT corpus in MERT training for phrase breaks has shown promising results, but they come with a caveat. Where does the SMT corpus come from? If we build a new SMT system, we can produce the text required to build a development corpus for MERT. However, it won't be labeled with phrase breaks. Because we don't have speech data corresponding to these translations, we won't have a way to automatically label the phrase breaks either. Thus, we would have to annotate a few hundred sentences for breaks by hand. It would be an interesting direction to explore how to reduce this effort. One particular idea is to study the divergence of phrasing across multiple SMT systems of different quality levels. There might be some generalization techniques that could be invented, that could allow MERT training for phrasing to be conducted on one well defined hand-labeled MT corpus, while the results still being valid for other MT outputs.

5.2.5 Phrasing and Video Dubbing

The number of video lectures available on the Internet today is increasing very rapidly. These lectures may be about topics of global interest, but are given in languages that may not be well understood everywhere. In interest of broader dissemination of knowledge, it is desirable to dub these lectures into several languages.

Video repositories such as TED have already taken steps to make their content widely accessible. Most videos are transcribed in their original language, and volunteers translate the text into several languages. Videos are then shown in the original language, but subtitles can be chosen in many languages. Other video hosting services, such as YouTube have automated some of these processes. Many English YouTube videos are automatically transcribed, and can be translated into one of many languages, enabling subtitles in a language of user's choice.

Our goal, via the PTSTAR project, funded by the Portuguese FCT, was to take a further leap in this direction. Instead of simply showing translated subtitles under original videos, our intention was to actually perform full speech to speech translation, and dub the video via speech synthesis in a foreign language of user's choice.

Automatic dubbing of video lectures is a tricky scenario. It consists of a speech recognition system that converts the original speech into text, and takes note of the timing of each text snippet. A machine translation system then converts this text into a different language. The speech synthesizer then takes the translation along with timing information and produces speech that is overlaid onto the video.

While at first inspection, video dubbing would appear to be the same problem as speech to speech translation, the "dubbing" mechanism makes it a trickier problem. When translating from one language into another, the length of underlying text usually changes. This is called the "fertility" of machine translation. The source speech could be be much shorter, or much longer that what needs to be produced in the target speech. However, because we have to overlay the new speech on top of the video, we would desire to achieve lip-sync. This would mean that depending on the difference in duration of source and target speech utterances for a particular snippet of a video, the speaking rate of the target speech would have to be shrunk or expanded. This can be done with the help of a global duration stretch over all phonetic segments. However, doing this has an impact on the intelligibility and naturalness of speech.

Our analysis and locally conducted experiments with five listeners showed that for a CLUSTERGEN voice, a duration stretch value between 0.7 and 1.4 was considered tolerable by listeners. This means, a ten second utterance could be sped up to 7 seconds, or slowed down to 14 seconds, without large impairment of intelligibility. We fixed the thresholds to be slightly more conservative than these, to aim for a duration stretch between 0.8 and 1.3 in the dubbing.

If an utterance required a duration stretch of say, 1.5, then putting a ceiling of 1.3 on our duration stretch factor would result in synthesis that is not in sync with the original video any more. In order to maintain the time sync, we used phrase breaks to our advantage. Let d_s be the duration of original speech, and d_t be the duration of synthesized translation, with normal synthesis. If $d_s = d_t$, then we can simply overlay the translation onto the original video. However, if $d_t < d_s$, then we need to apply a duration stretch factor of $S = d_s/d_t$ to the synthesis. If $S < 1.3$, then we use the required stretch factor and perform the synthesis. However, if $S >= 1.3$, then we use a stretch factor of 1.3 for all segments other than pauses, and the remainder time is covered by expanding the duration of pauses. A duration stretch limit of 1.5 was used for pauses within an utterance, and pauses at the beginning and end of the utterance were used to cover for the remainder time. In the event that $d_t > d_s$, the stretch factor would be $S < 1$. If this was within the limits of $S >= 0.8$, we used the required stretch. However, if $S < 0.8$, then we set the stretch to be exactly 0.8, and let the synthesis spill over. The extra time was then carried over to the next segment and the stretch factor of that segment was adjusted to fit the actual remaining time (after the spillover from previous sentence).

The algorithm described here seemed to work well. We used the TED video database to perform automatic dubbing, and found that without a significant loss in intelligibility, dubbing could be achieved with very good lip sync. We could not formally evaluate how well the sync actually was, because it is difficult to define what the gold standard should be in this case. It is also difficult to design a subjective test that can allow people to critique the achieved sync.

The method described here for video dubbing is essentially heuristic. However, there is scope for future improvement here with the help of methods proposed in this thesis. For example, we only varied the duration of phrase breaks in our method. However, the phrasing "knob" could be used to vary the number of phrase breaks generated thereby allowing us to reduce the duration stretch even further. The current speech translation framework however limits us from testing the knob approach. The speech recognition and translation work at the level of few words at a time, and do not take full utterances (or sentences). This is not an ideal setting for the phrasing model, and especially the knob, because it would result in breaks that are too frequent. With an improved speech translation framework available for video dubbing, the phrasing knob could be incorporated into the video dubbing pipeline.

5.3 Chapter Summary

In this chapter, we studied the impact that phrasing can have on intonational models, and its effect on intelligibility of synthesized automatic translations. We saw that a better phrasing model can help make the pitch prediction be more natural. We then saw that phrasing can improve the understandability of disfluent sentences. We saw how the MERT framework proposed in this thesis can adapt itself to the task of machine translation. We also described the task of video dubbing and how our heuristic methods in achieving time synchronization between original and translated speech.

Phrasing can also have an impact on the segmental duration. In fact, duration modeling happens between phrasing and pitch prediction, and thus an improved duration model could enhance the intonation model even further. This thesis has not looked at the impact of phrasing on segmental duration, or the cascaded effect on pitch prediction.

We have only scratched the surface here, of prosody in speech to speech translation. Phrase breaks are used an intelligibility enhancement devices, to cover up mistakes that speech recognition and machine translation could have introduced. However, a deeper integration of the recognition, translation and synthesis components would be a valuable endeavor: prosodic information such as phrase breaks, intonation and duration could be transferred from the source side into the target side leading to a more appealing spoken translation.

Parts of work presented in this chapter have been published in (Parlikar et al., 2010) and in annual reports of the PT-STAR project.

Chapter 6

Phrasing for Low Resource Languages

When words are scarce they are
seldom spent in vain.

William Shakespeare

Languages of the world can be divided into three zones. The "green" zone of languages contains languages like English, that are well studied, have a lot of speakers and researchers. Such languages typically have not just lot of data available but also a wide variety of linguistic tools to process the language. These tools could be lexical (dictionaries), syntactic (parsers), semantic (thesauri), or even higher level such as discourse interpreters. Very few languages of the world could fall into the "green" zone. Many languages belong to the "yellow" zone. This is where data might be easy to find, and the language may be well known, but linguistic processing tools are not easily available. However, most languages (and their dialects) end up in a "red" zone. Neither data, nor tools are easily available and in many cases, linguistic background of the language is also difficult to find.

From a speech synthesis perspective, in terms of building voices, we can redefine these zones as follows: the green zone has languages like English, where speech data is available with transcripts, and linguistic tools such as part-of-speech taggers and parsers are available. The yellow zone has languages for which speech data and transcripts are available, but no text-processing front end, or linguistic tools are available. The red zone is languages where only speech data is available. Transcripts are either not available because nobody produced them, or because the language does not have a standardized writing system.

Speech synthesizers need to be deployed for all the three zones of languages. Most of the work we presented so far has catered to the green zone languages. Very little work has been done so far to predict phrase breaks when working with the yellow and red zone languages. The most common solution is to use a punctuation-based phrasing model: where there is a punctuation, there is a break. However, we would like to extend our data-driven methods to also work for these yellow and red zone languages.

We shall first look at the yellow zone, the low-resource languages and study how our method can be extended to build data-driven grammar-based phrasing models with the help of automatic part-of-speech induction. We shall then look at the red zone, describe how we build voices for such languages, and investigate whether phrasing models are feasible in such scenarios.

6.1 Handling Low-Resource Languages

We often build voices for languages other than English. While Festival has a sophisticated default phrasing model for English, the default for other languages is simply the punctuation based model and can be quite unreliable. This is because not all languages use punctuation to denote phrase breaks as in English, and also because some genres of text are well punctuated, whereas others may not be punctuated at all. While capturing stylistic phrasing was the primary objective of our modeling strategies, we investigated if our models could be adapted to handle new languages.

Our grammar based model depends on part of speech tags to predict phrase breaks. These tags are required both as lexical features in the decision tree, but also as terminal symbols of the grammar we train. If we are dealing with a new language, we can either use a tagger in that language if we have one. Otherwise, we can perform automatic induction of POS tags. To do so, we use the Ney et al. (1994) clustering algorithm as implemented by Clark (2003). This algorithm iteratively improves the likelihood of a given clustering by moving each word from its current cluster to a cluster that will maximize the increase in likelihood. We only cluster words that appeared in our corpus over 1000 times, and group them into 16 clusters. We use these clusters without any manual modifications and use the cluster numbers as POS categories. To distinguish these tags from actual POS tags, we shall call tags induced with this method as IPOS tags. At test time, if we encounter a word for which an IPOS tag was not found, we set a default tag called *content*. In this section, we shall look at how well our method does in the low-resource scenario.

6.1.1 Languages and Available Resources

We wanted to carry out experiments on languages that differ in families as well as amount of linguistic resources available. We chose English, European Portuguese and Marathi for this work. English is a Germanic language with rich set of linguistic tools. Portuguese is a Romance language and has many linguistic resources available in general. Marathi is an Indo-Aryan language spoken in India, and all we had access to was a text corpus.

Our English voice was trained on the F2B corpus (about 55 mins of speech) from the Boston University Radio News Corpus (Ostendorf et al., 1995). We used the English POS tagger available within Festival. We also induced IPOS tags over 50000 sentences taken from the English side of the Europarl (Koehn, 2005) corpus. For running listening tests, we randomly selected 25 long utterances from the F1A set of the Boston University Radio News Corpus.

We built our Portuguese voice from about an hour of speech of recordings of a male news broadcaster from Portuguese national TV. We did not have access to a Portuguese POS tagger. We did have a lexicon that provides part of speech for known words, but does not disambiguate multiple possible POS based on context. We used 50000 sentences from the English-Portuguese Europarl corpus and induced IPOS tags for Portuguese.

For running listening tests, we selected 15 long utterances from online Portuguese newspapers.

We only had a text corpus available for Marathi. This was a collection of news published in the E-Sakal newspaper. The corpus was collected at the Center for Indian Language Technology at IIT Bombay. We had about about half an hour of speech recorded by the AUP voice in order to build a synthetic voice, and build phrasing models. There was no POS tagger or lexicon available. We used 50000 sentences from the text corpus to induce IPOS tags for Marathi. For listening tests, we selected 15 long utterances from this same text corpus.

Phrase prediction is an easier problem when text is well punctuated. In order to simulate the harder (and the more important) case when punctuation is not available to us during synthesis, we stripped all our corpora for punctuation within utterances for all languages. We let the sentence final punctuation remain in text.

Note that for all three languages, we ran IPOS learning only on 50000 sentences. We did have access to much larger text corpus in all languages, but we decided to using a corpus of this size to make sure our technique works when for a new language we might not have hundreds of thousands of lines of text available.

6.1.2 Experimental Results

We have several phrasing models to compare in this experiment. For all languages, our baseline model is the punctuation-based model. However, because the text we were synthesizing did not have punctuation, there was no phrasing being done. We call this model the NONE model.

Evaluation for this section uses the F-measure and the L2 and EMD distances presented in Chapter 2. We also show results of subjective listening tests run on all three languages. For English, we used Amazon Mechanical Turk (MTurk) to run the listening task. We split 25 utterances into sets of 5x5. Each set was presented as an individual HIT. We allowed 10 workers per HIT. Thus, we had 50 tasks, and 5 utterances each, giving us 250 data points for comparison. We discarded responses by few workers on MTurk since they had finished the task too quickly, and their responses would have been spam. For Portuguese and Marathi we could not reliably use MTurk for the listening task. We requested volunteer native speakers of the languages to perform the task. Majority of our Portuguese participants did the task over the web from Portugal, and similarly majority Marathi tests were taken in India. We had about 100 data points for comparison for Portuguese experiments, and 120 data points for Marathi. After collecting data of the subjective task, we simply counted the total percentage of votes received by each model in an experiment. The model that receives the majority vote can be thought of as the winning model.

English had the most resources available among the three languages. We also had the default Festival phrasing model available here. We thus used English to perform "oracle" study and run sanity checks to ensure that our approach is indeed moving in the right direction. We have four different phrasing models for English to compare: (i) The NONE model, (ii) Festival's default model, (iii) Grammar based phrasing using Festival's POS tags, and (iv) Grammar based phrasing using the IPOS tags. Table 6.1 shows the results of objective evaluation of these four models. The results presented here are the average values after performing 10-fold cross validation.

Table 6.1: Objective Results for Phrasing in English

System	F1	L2	EMD
NONE	0.0000	0.2566	10.6233
Festival	0.3417	0.2802	3.0733
POS Phrasing	0.3481	0.1661	1.1449
IPOS Phrasing	0.2751	0.1972	1.7744

Based on the results in Table 6.1 and performing significance analysis, we can draw the following conclusions for p-value $p < 0.01$:

- Grammar based POS phrasing model is slightly better than the default model in Festival. The improvement in F-1 measure is not significant, but the improvement in L2 and EMD measures is significant.

- Grammar based IPOS phrasing model is slightly weaker than the Grammar based POS model across all metrics, but the differences are not statistically significant.

- Both the IPOS and POS models are significantly better than the NONE model.

We wanted to see if subjective listening tests support the objective comparisons here. We did two listening tests. First, we compared the NONE model to the IPOS model. Figure 6.1 shows the results for this. We found that the IPOS model is better than the NONE model. The result is statistically significant.

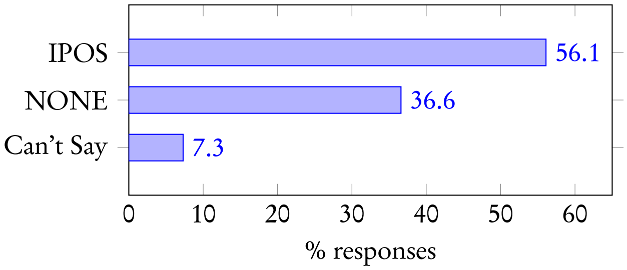


Figure 6.1: English Subjective Test: Which model is better?

In the second test, we compared the IPOS model to the POS model. Figure 6.2 shows these results. We found that while the POS model gets more votes overall compared to the IPOS model, the difference is not statistically significant.

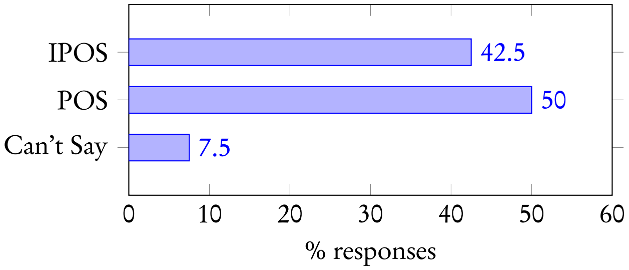


Figure 6.2: English Subjective Test: Which model is better?

The objective and subjective results on English show that that using the Grammar based approach with POS tags helps us do better at phrasing than the standard model in Festival. We also see that replacing the POS tagger with IPOS tags also gives us a very reasonable phrasing model.

For Portuguese, we only have three phrasing models: (i) The NONE model, (ii) The Grammar based POS model, and (iii) The Grammar based POS model. Note that the POS model here is slightly different than the one available for English, because we only had a lexical part of speech available for Portuguese. Table 6.2 summarizes the objective results for Portuguese phrasing. The results presented here are average values after performing 10 fold cross validation.

Table 6.2: Objective Results for Phrasing in Portuguese

System	F1	L2	EMD
NONE	0.0000	0.4113	28.2284
IPOS Phrasing	0.2870	0.2427	2.9735
POS Phrasing	0.2520	0.2639	3.2327

After performing significance analysis over these objective results, we found like just like for English, we could make the following conclusions:

- Both the IPOS and POS models are significantly better than the NONE model.
- The IPOS model is not significantly different compared to the POS model.

We tried to verify with listening tests, whether these hypotheses hold true for subjective opinion also. We did two listening tests, similar to those in English.

In the first listening test, we compared the NONE model to the IPOS model. Figure 6.3 shows this result. In the second test, we compared the IPOS model to the POS model. Figure 6.4 shows this result. Numerically, we see that the IPOS model is better than the NONE model, and that the POS model is better than the IPOS model. However, Significance analysis showed that the three systems may not be significantly different on the listening tasks.

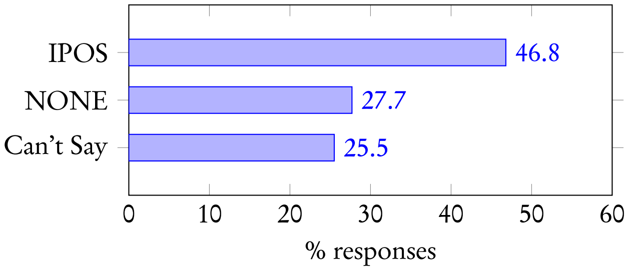


Figure 6.3: Portuguese Subjective Test: Which model is better?

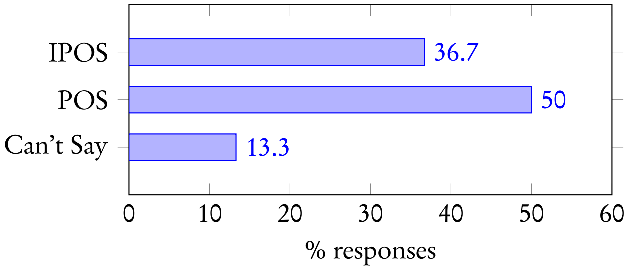


Figure 6.4: Portuguese Subjective Test: Which model is better?

We only have two phrasing models for Marathi: (i) The NONE model, and (ii) A Grammar based model trained with IPOS tags. Table 6.3 summarizes the average objective results after 10 fold cross validation, comparing these two models, and subjective results are presented in Figure 6.5. We see that both objectively and subjectively, the IPOS model is significantly ($p < 0.01$) better than not having phrasing at all.

Table 6.3: Objective Results for Phrasing in Marathi

System	F1	L2	EMD
NONE	0.0000	0.1850	2.1491
IPOS Phrasing	0.2560	0.1828	0.8352

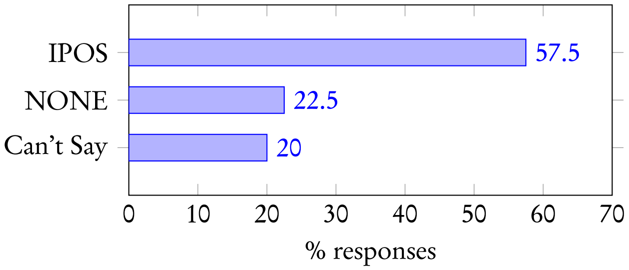


Figure 6.5: Marathi Subjective Test: Which model is better?

6.1.3 Summary: Phrasing for Low Resource Languages

In this section on phrasing for low resource languages, we have shown that our proposed grammar, together with automatic POS induction can help build phrasing models for languages that do not have very many resources. In the case of Marathi, we built both the voice and the phrasing models using only half an hour of speech. While we do need a few thousand lines of text corpus to induce POS tags over, such corpus can be easy to find. Our English results in Table 6.1 show that given the lack of punctuation in input text, our proposed approach works better than Festival's default models with POS tags coming from the tagger. If we use IPOS tags, then the performance goes down slightly. This suggests that adding more linguistic information certainly helps. However, the difference between models built using the POS versus IPOS tags is not very high, and hence investing time and money in procuring linguistic resources may not always be necessary.

We primarily dealt with part-of-speech induction in this part of the work, and did not look at other linguistic analysis, such as morphology, parsing and semantics. Vadapalli et al. (2013) have recently suggested that word-terminal syllables could be used as good indicators of phrasing for Indian languages. Using last and penultimate syllables as features might also work for other languages that use morphological case markings. The induced parts of speech we used are an approximation of lexical parts of speech, when taggers are not available. In the context of low-resource languages, there is need for research in two directions: (i) Developing heuristic or automatic techniques that approximate linguistic analysis, and

(ii) Investigating how these developments could be used in improving the phrasing, and more generally, speech synthesis.

6.2 Enabling TTS without Text

Many languages of the world, and most dialects do not have a standardized writing system, yet are spoken by many people. If speech technology is to make an impact for all languages, it will need to consider the processing of languages without a standardized orthography. We have recently begun the investigation of building text to speech systems for languages where a text form isn't available. We expect to be able to collect acoustics in that language, and be able to know the meaning of what is said. Given only speech data in that language that we can use at training time, we want to build a text to speech system.

At first it may seem futile to develop a speech synthesis system without a related writing system. But consider these two use cases that highlight the need of such a system. The first is a speech translation system, that takes content spoken in a language like English, and needs to be "dubbed" or translated into a language that only has a spoken form. A second use case is deployment of spoken dialog systems in the language that has no written form.

If text is fundamental to "text-to-speech", what does it even mean to build speech synthesizers for languages where text is not available? Our proposal is that we can use the speech corpus to automatically derive a written form for that language. This could be in the form of a phonetic writing system that uses either an universal phone set, or a phone set from a closely related language. A cross-lingual phonetic decoder can be used to automatically derive such a written form for our target language. Once this automatic written form is available, we can train a speech translation system, or a dialog system's natural language generation unit to produce text in this form. The synthetic voice we train will then be able to process this text and produce speech in the target language.

6.2.1 Crosslingual Phonetic Decoding

We have developed an iterative cross-lingual phonetic decoding method that allows us to build text-to-speech without text. We have shown, in (Palkar et al., 2012; Sitaram et al., 2013b,a) that this method works for several languages. Figure 6.2.1 shows the overview of our method.

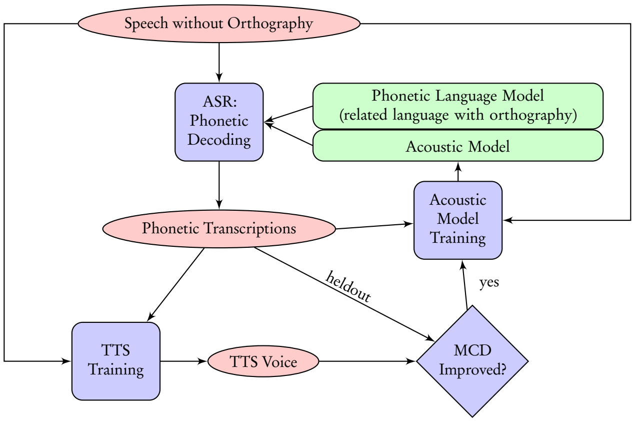


Figure 6.6: Overview of our method to build TTS voices from speech data without transcripts

We start by building a cross-lingual phonetic decoder. The acoustic model is built on a high resource language, ideally with a phone set that is close to the target language. The language model is built over phone sequences of a language that is a high resource language also phonetically close to the target language. We then use the speech corpus we have and decode it with the phonetic decoder to obtain phonetic transcriptions. We now start an iterative process of adapting the acoustic model to fit to the data we have. We take the target speech and its obtained transcripts and build a new acoustic model. We decode the speech again with the new acoustic model, keeping the language model the same, and produce new transcripts. We retrain a new acoustic model with the new transcripts, and repeat the decoding over many iterations. At each iteration, we build a CLUSTERGEN voice and measure its spectral quality in terms of the MCD distance. Because this early work was exploratory in nature, we did some experiments with languages that do have an orthography, but pretended they didn't. Figure 6.2.1 shows a graph of how the MCD improves over different iterations for German and for English. For German, we used the WSJ acoustic English acoustic model, and a 3-gram language model defined over English phones learned from the Europarl corpus. For English, we used an acoustic model trained over the Indic database by Prahallad et al. (2012) and trained a 3-gram language model over German phones taken from the Europarl corpus. We see from the figure that while the improvement in the MCD is not monotonic, many of the iterations are better than the baseline crosslingual phonetic decoding we start with.

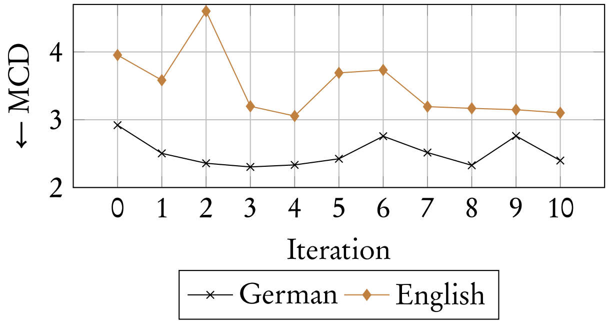


Figure 6.7: MCD of voices built from transcriptions for German and English over the iterative decoding process

6.2.2 Phrasing, Prosody, Word Discovery

Our initial experiments and subjective listening tests so far show that the spectral quality of the speech we produce is understandable. However, the prosody is far from being acceptable. We would want to build phrasing models, and other prosodic models for these voices.

One reason why the prosody of voices we have built is not very good is that these are phonetic voices: something speech synthesis is not designed to deal with. Phrasing, duration modeling, and intonational models all use word-level information, such as lengths of words, position of a segment or syllable in a word, etc. as features in their prediction task. By synthesizing purely from phoneme sequences, this critical information is not available.

Since the focus of this thesis is on phrasing, we looked at what the impact of losing the high level word-information is on the phrasing models.

We did an oracle experiment with the F2B corpus. We built three voices with this corpus: one where the text was words, one where the text was strings of syllables, and one where the text was strings of phonemes. We processed the word-level corpus here to strip punctuation, and as such the results here are not comparable to other phrasing results on the F2B corpus published in this thesis.

We induced part of speech tags over the three corpora and built grammar based phrasing models. We then measured the performance of those models over held-out data by looking at the F-measure of the prediction. Table 6.4 shows these results. We see that the loss in phrasing accuracy is very high when moving from a word level to the syllable level, and even more so when we synthesize from text that contains strings of phones.

Table 6.4: F-measure of phrasing models trained over text at different types of units

Text Unit	F1
Words	21.62
Syllables	12.28
Phonemes	7.80

We thus have three things: (i) The voice we are building is a phonetic voice, (ii) Phrasing model is very weak at the phonetic level, and (iii) We need a good phrasing model to improve prosody in the voice we have. There are two potential solutions to address this situation in the future: (i) Develop strategies to build a phrasing model at the phonetic level, and (ii) Automatically discover word-like units from phonetic segments in the derived orthography.

Phrase breaks can be of help in discovering words automatically: the data-driven method we have used through out this thesis labels position of phrase breaks from speech data. If we have a string of phonemes with the associated labels of where the phrase breaks are, we know some of the word boundaries. This constraint can possibly improve the performance of word induction algorithms.

6.3 Chapter Summary

In this chapter, we looked at phrasing and its performance on low-resource languages. We showed how we can used part of speech induction to enable our grammar-based phrasing model to work with languages that do not have part of speech taggers available. Since this method is language independent, we can use it for any language, instead of the commonly used "punctuation" rule for phrasing. This work has been published in (Parlikar and Black, 2012a).

We also looked at our team effort on building text to speech systems for languages without an orthography. We saw that we can build phonetic voices that are of an acceptable spectral quality, but lack quite a bit in prosody. We saw that not having word-level units in text can severely degrade the performance of our phrasing model (which has been optimized for words). We hope that improvements in word discovery algorithm from phoneme strings can help improve the prosody in this setting.

Chapter 7

Conclusions and Future Work

> In literature and in life we ultimately pursue, not conclusions, but beginnings.
>
> Sam Tanenhaus

This thesis has looked into speaking styles and addressed the problem of phrase break prediction. The presented methods lay down a foundation of stylistic synthetic prosody. Other prosody models can benefit from this foundation. Together, this can help making parametric speech synthesis sound more natural, as well has have an impact on applications of speech synthesis such as speech to speech translation.

We have shown that speaking styles vary both in the placement and duration of phrase breaks within an utterance. We have also shown that people can perceive differences between different placements and different durations. Modeling phrase breaks appropriately is therefore important for perceptually better synthesis. We have shown that the proposed Grammar-Based method and the minimum error rate framework allow us to capture stylistic phrasing in a data-driven setting. We have also shown that the methods are extensible to low-resource scenarios, and that the techniques can be applied to uncommon languages. Traditionally, phrasing models have been trained on large hand-annotated corpora. The proposed data-driven language-independent methods to manipulate placement, duration and rate of phrase breaks is thus a novel contribution.

This thesis proposed the grammar-based method and showed that it makes phrasing models better. However, the method itself, along with its language-independent extensions are generic and can be applied to problems other than phrasing. Indeed, this technique has been directly used by Anumanchipalli (2013) for predicting accent groups in intonation modeling. The proposed Minimum Error Rate Training framework is also very extensible. While we have only seen its impact on phrasing in this thesis, the technique and code is designed to be useful in other modeling methods where optimization towards an error metric is desirable.

TestVox, the web based subjective evaluation framework is also an important contribution of the thesis. It simplifies the process of setting up listening tests, and allows tests to be deployed to crowd-sourcing platforms. We hope this tool will be of great value for future research in speech synthesis.

7.1 Future Directions

This thesis has looked at only one particular type of phrase breaks: the pauses within utterances. These correspond to the TOBI level 3. We have not directly addressed the level 2 and level 4 breaks. Once accurate methods for data-driven labeling of tonal breaks (level 2) from speech corpora are developed, those annotations can be directly used in the architectures proposed in this thesis and modeled in conjunction with the Anumanchipalli et al. (2011) model to generate synthetic tonal breaks. Major breaks (level 4) happen at ends of utterances. While the placement of end-of-utterance breaks is trivial, a lot can be done to model duration and intonation at these boundaries.

Speech synthesizers typically synthesize isolated utterances. However, the context of a sentence has an impact on the boundary between two utterances. This context can affect how much pause there is between the sentences, and how pitch resets for the next utterance. Hovy et al. (2013) describe how people speak differently when reading isolated sentences versus sentences spoken in context. Modeling these cross-sentential aspects of pitch and duration can make the speech sound more natural. Features that could help us model cross-sentence events could be at the lexical level, such as overlap of content words between the two utterances, or the lengths of the utterances. However a deeper analysis of what is happening at the boundary could be more helpful: whether a narrative is continuing, or whether a quotation begins or ends, or whether new information is being presented, and whether entities in the next utterance have first or second mentions. It might be that the overall discourse of current speech has an impact on our utterances and the boundaries between utterances. However, we still need to investigate how far back in the discourse we should extract features from, for our modeling purpose. Considering a couple of utterances around the boundary is practical, going beyond paragraphs, or to the chapter level might cause us to hit data sparsity issues. There also is an cognitive aspect to be investigated here: a speech synthesizer can use "future" utterances to predict events at an utterance boundary. However, when we speak, we typically do not construct utterances well in advance. In that case, is the information contained in future utterances helpful for modeling inter-utterance events? And if so, how does the human brain use this "future" information, i.e., in what form could we pre-generate the speech we have to soon produce, and how far a look-ahead does the human brain do?

Our proposal of the grammar-based phrasing method uses the formalism of a context-free grammar. While our results show that this is useful for capturing acoustic syntax, it may be possible that this formalism is not ideal. Linguistic studies have focused on syntax, such as found in the Penn Tree Bank (Marcus et al., 1994). But acoustic syntax is not yet formally studied. It may be possible to come up with a grammar formalism that can help chunk text into its acoustic parse in a manner superior to what we have used in this work.

In this work, we used syntactic features with the grammar based model. The next step is to see how semantic features could help make further improvements. Our experiments with phrasing in the context of speech translation tell us that appropriate phrase breaks can help make speech more understandable. The same might be true for enhancing the intelligibility of semantically complex content. If an utterance contains hard words, or rare words, pauses could help make them more intelligible. The use of a language model in the MERT framework is a step in this direction, as rare words will have a lower probability. However, the language model can at best be considered to be a shallow semantic model. Semantic analysis of the utterance we want to speak, in terms of the entities and relationships in the utterance could provide additional features for MERT and have an impact on the pauses. It would be interesting to see how semantic features can be added to the MERT framework and what impact it would have on the phrasing accuracy.

Phrasing is the foundation of synthetic prosody, and this thesis addresses the problem of inserting appropriate phrase breaks. Work by Anumanchipalli (2013) addresses the problem of capturing style-specific intonation. However, style-specific models to capture segmental duration need to be thoroughly investigated now. In particular, it would be interesting to explore whether, instead of a cascade of phrasing, duration and intonation models, we could jointly make decisions about the individual models in order to make the overall prosody more natural.

This work has used subjective evaluations to make modeling decisions, as well as to show how methods compare to one another. However, the question we have most typically asked in an A/B test is for participants to tell us which model they prefer. While this test seems to be valid for comparing different phrasing models, it is not perfect. Given that statistical parametric synthesis has now reached levels of adequate naturalness and expressiveness, there is imminent need for a full scale cognitive study on how to design experiments for subjective evaluation of speech synthesis. There is also a need to study how these subjective tests correlate with different objective tests, so that researchers can optimize their models to the appropriate objective criterion.

One way of designing subjective evaluation for speech synthesis is to build it around the exact application that a synthesizer is being designed for. For example, if the synthesizer will be used as the speech interface of a dialog system, then subjective tests should measure the impact of our models (phrasing, or others) in terms of the task completion rate, or task completion time. Skantze et al. (2013) have shown that users' behavior is affected by how pauses are realized in the speech of a dialog system. Pauses can have an effect on turn taking, grounding, and explicit and implicit confirmation. The duration and placement of a pause can help people detect whether the floor is theirs to take, and a user study could be conducted to see whether people react appropriately to the generated pauses. Pauses can help with implicit confirmation: if a system accepts user input, repeats it as a declarative sentence ("Going to Pittsburgh.") and pauses for a while, users get an opportunity to barge in and correct the system before it moves to the next dialog act. The duration of the pause could be linked to the system's confidence in the recognition of user input. These can be used to design user study that evaluates how well our speech synthesizer and prosody models are doing. In other dialog applications, such as spoken directions and navigation, the speech synthesizer could be evaluated based by gaze tracking, to see if users understood the speech by itself or whether they had to look at their navigators in addition. A synthetic voice that minimizes driver's distraction from the roads could be deemed to be better.

If subjective evaluation of speech synthesis is based on task completion, then so should be the modeling. Prosody models in a dialog system, for example, could be built conditioned on the dialog state of the system. If the system is confirming user's input for example, the prosody could be quite different, compared to when the system is offering choice to the user. In a navigation scenario, spoken directions could be given differently based on the cognitive load on the driver—the traffic conditions, the complexity of the route, and the familiarity of the driver with the area. Training these models will be a move from the general "style-specific" approach to a "task-specific" approach.

We used a data-driven definition of speaking style in this work. By using recorded corpora as our gold standard for phrasing, we defined our models to capture the style found in the corpus. However, style is a function of various parameters. It is very difficult to tease apart aspects of speaking style that are speaker-specific, task-specific, and genre-specific. A more detailed analysis of speaking styles, within the conform of this data-driven definition is necessary in order to understand how to tease these aspects apart when building our models.

This thesis has barely touched the surface of modeling prosody in the speech to speech translation task. Phrasing in the target side was predicted purely from the target text, with the help of a language model.

This may capture events such as bad word combinations, and improve the overall intelligibility. However, there is no direct connection between the synthesized phrase breaks and the breaks in original speech. Phrase breaks can be used for effects of emphasis or dramatics, and that information is so far lost in translation. There is a need for deeper integration of the speech recognition, machine translation and text to speech systems in order to transfer all aspects of prosody from the source side into the target side.

APPENDIX A

TestVox: Web Based Subjective Evaluation

O Wad some Pow'r the giftie gie us
To see oursels as others see us!

Robert Burns

Speech synthesis is an end-user technology. The output of a speech synthesizer is usually consumed by people. Very rarely is it fed to other speech and language technologies for further processing. This means that any improvements that we think we made to our speech synthesis models need to be validated with the help of listening tests.

Subjective evaluation is integral to speech synthesis research, yet the process of setting up and performing listening tests is challenging. Traditionally, tests were set up in a lab environment, with controlled equipment, and volunteers or paid participants were recruited to listen to stimuli and submit responses. However, research activities might need a lot of tests, frequently paced, and finding participants for listening experiments might be difficult. In fact, conducting tests locally in lab settings might even be impossible for some smaller labs in remote locations.

Internet has become a popular platform for conducting listening tests. While this enables remote subjects to participate in the tasks, experimenters lose control over the listening equipment that is used at the remote end. The biggest advantage, however, is that a lot of people can be recruited for tests, and tests can be completed faster. Platforms such as Amazon Mechanical Turk have even enabled paying such subjects in a convenient manner. In a way, speech synthesis evaluation was always a 'crowd-sourced' process. But official crowd-sourcing platforms, such as Amazon's Turk have made the process very convenient.

Although the web has come out as the biggest solution to fast, cheap, and reliable listening tests, the biggest problem of running tests on the web is web standards. Different browsers, different operating systems often have infamously disparate implementations of identical technologies. This is especially true of how audio can be played on the web. Getting speech produced appropriately to users with every and all browsers that are commonly used can be very tricky. The other problem with hosting tests for the web is that many smaller research teams don't necessarily have access to a public IP address on which they can host tests. Requiring to find a hosting provider, and then setting up web-servers and relevant tools to conduct listening tests is complicated, and requires a lot of investment (time, money) on part of researchers.

Many technical contributions of this thesis had to be validated using listening tests, and over the course of running these experiments, we have developed a solution called TestVox, which simplifies running listening tests. TestVox is publicly released software under an open-source license, and is an important contribution of this thesis work.

A.1 Features

TestVox is a web-based framework for subjective evaluation of speech synthesis. It helps quickly setup and run listening tests over the Internet. It supports many of the commonly used listening test formats (A/B tests, MOS tests, Transcription tests, Word-Choice tests, etc.) It supports entry and exit surveys. Audio stimuli can be presented in random order if desired. TestVox is designed to be cross-browser compatible. It can run under Linux, Windows or OSX. It could also be deployed to an instance of the Google App Engine computing environment, which can be useful if a public IP address is not directly available to researchers.

TestVox comes with a built-in webserver, so there is very little to configure and setup. It can also be run under an existing server such as Apache or nginX. TestVox comes with scripts that simplify setting up listening tests by taking wav files as input and setting up a zip file that can be uploaded to an admin interface to start listening tests. Once participants submit responses, the answers can be retrieved as a CSV file to analyze and summarize results. The Mechanical Turk interface for TestVox is also powerful. It allows posting, listing and reviewing tasks easily.

The motivation behind TestVox is not just to simplify uploading and posting listening tests, but also create recipes for analyzing and summarizing results. Crowd-sourced listening tests often have to be filtered for spam responses. TestVox aims to create recipes for inculcating best practices for such filtering in standardized analysis scripts that would allow researchers to quickly evaluate their models.

A.2 About

TestVox is primarily written in Python. It has about 4.5k lines of code, and is estimated to be about an year of effort under the COCOMO model. It consists of a server, that runs under the control of speech researchers, and a web-client that participants use to listen to stimuli and submit responses. The web-client is written with the help of PyJS. The TestVox framework, tools and documentation are licensed under a BSD like license, free to use without restriction (commercial or otherwise). More information is available on the TestVox website.

Bibliography

Jordi Adell, David Escudero, and Antonio Bonafonte.
Production of filled pauses in concatenative speech synthesis based on the underlying fluent sentence.
Speech Communication, 54(3):459–476, 2012.
Cited in: 5.2.1

Pablo Daniel Agüero, Jordi Adell, and Antonio Bonafonte.
Prosody Generation for Speech-to-Speech Translation.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006.
Cited in: 5.2

Gopala Krishna Anumanchipalli.
Intra-lingual and Cross-lingual Prosody Modelling.
PhD thesis, Carnegie Mellon University, 2013.
Cited in: 2.5, 5.1, 7, 7.1

Gopala Krishna Anumanchipalli, Luis C. Oliveira, and Alan W Black.
A Statistical Phrase/Accent Model for Intonation Modeling.
In *Proceedings of Interspeech*, pages 1813–1816, Florence, Italy, August 2011.
Cited in: 2, 5.1, 7.1

G. M. Ayers.
Discourse functions of pitch range in spontaneous and read speech.
Working papers in linguistics, 44:1–49, 1994.
Cited in: 1.1

J. Bachenko, E. Fitzpatrick, and C. Wright.
The Contribution of Parsing to Prosodic Phrasing in an Experimental Text-to-speech system.
In *Proceedings of Association for Computational Linguistics*, pages 145–153, New York, New York, 1986.
Cited in: 2.1

J. Bachenko, E. Fitzpatrick, and C. Wright.

A computational grammar of discourse-neutral prosodic phrasing in English.
Computational Linguistics, 16:155–170, September 1990.
URL `http://dl.acm.org/citation.cfm?id=98377.98380`.
Cited in: 2.1

Satanjeev Banerjee and Alon Lavie.
METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.
In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
Cited in: 5.2.2

Plínio Almeida Barbosa and Gérard Bailly.
Progress in speech synthesis, chapter Generation of pauses within the z-score model, pages 365–381.
Springer Verlag, New York, 1997.
Cited in: 4

William J. Barry.
Phonetics and phonology of speaking styles.
In *Proceedings of 13th International Congress of Phonetic Sciences*, volume 2, pages 4–10, Stockholm, August 1995.
Cited in: 1.1

Peter Bell, Tina Burrows, and Paul Taylor.
Adaptation of Prosodic Phrasing Models.
In *Proceedings of the 3rd International Conference on Speech Prosody*, Dresden, Germany, May 2006.
Cited in: 2.1

Tatiana Benaglia, Didier Chauveau, David R. Hunter, and Derek Young.
mixtools: An R Package for Analyzing Finite Mixture Models.
Journal of Statistical Software, 32(6):1–29, 2009.
URL `http://www.jstatsoft.org/v32/i06/`.
Cited in: 4.2

Nicola Bertoldi, Richard Zens, and Marcello Federico.
Speech Translation by Confusion Network Decoding.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 1297–1300, Honolulu, HI, USA, April 2007.
Cited in: 5.2

Eleonora Blaauw.
Phonetic differences between read and spontaneous speech.
In *Proceedings of 2nd International Conference on Spoken Language Processing*, pages 751–754, Banff, Alberta, Canada, October 1992.

Cited in: 1.1

Alan W Black.
CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling.
In *Proceedings of Interspeech*, pages 194–197, Pittsburgh, Pennsylvania, September 2006.
Cited in: 3.3

Alan W Black and Paul Taylor.
The Festival Speech Synthesis System: system documentation.
Technical report, Human Communication Research Centre, University of Edinburgh, January 1997.
URL `http://www.cstr.ed.ac.uk/projects/festival`.
Cited in: 1.5, 2, 2.3, 4

Alan W Black and Keiichi Tokuda.
Blizzard Challenge – 2005: Evaluating corpus-based speech synthesis on common datasets.
In *Interspeech*, pages 77–80, Lisbon, Portugal, September 2005.
Cited in: 1

Anthony Bladon, Rolf Carlson, Björn Granström, Sheri Hunnicutt, and Inger Karlsson.
A text-to-speech system for British English, and issues of dialect and style.
In *ECST*, pages 1055–1058, 1987.
Cited in: 1.1

Antonio Bonafonte, Pablo D. Agüero, Jordi Adell, Javier Pérez, and Asuncioń Moreno.
Ogmios: The UPC text-to-speech synthesis system for spoken translation.
In *Proceedings of the TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, June 2006.
Cited in: 5.2.1

Catherine P Browman and Louis Goldstein.
Tiers in articulatory phonology, with some implications for casual speech.
Papers in Laboratory Phonology I: Between the grammar and physics of speech, pages 341–376, 1990.
Cited in: 1.1

Gösta Bruce.
Modelling Swedish intonation for read and spontaneous speech.
In *Proceedings of International Congress on Phonetic Sciences*, volume 2, pages 28–35, 1995.
Cited in: 1.1

Peter Cahill, Jinhua Du, Andy Way, and Julie Carson-Berndsen.
Using Same-Language Machine Translation to Create Alternative Target Sequences for Text-to-Speech Synthesis.

In *Proceedings of Interspeech*, pages 1307–1310, Brighton, United Kingdom, September 2009.
Cited in: 5.2.1, 5.2.2

Wilhelm Nicholas Campbell.
Multi-level Timing in Speech.
PhD thesis, Sussex University, U.K. Department of Experimental Psychology., 1992.
Cited in: 4

Estelle Campione and Jean Véronis.
A large-scale multilingual study of silent pause duration.
In *Proceedings of the 1st International Conference on Speech Prosody*, pages 199–202, Aix-en-Provence, France, April 2002.
Cited in: 4.2

Francine Robina Chen.
Acoustic Characteristics and Intelligibility of Clear and Conversational Speech at the Segmental Level.
Master's thesis, Massachusetts Institute of Technology, 1980.
Cited in: 1.1

Alexander Clark.
Combining distributional and morphological information for part of speech induction.
In *Proceedings of European Chapter of Association for Computational Linguistics*, pages 59–66, Budapest, Hungary, August 2003.
Cited in: 6.1

Jacob Cohen.
A Coefficient of Agreement for Nominal Scales.
Educational and Psychological Measurement, 20(1):37–46, April 1960.
Cited in: 5.2.3

Scott Cohen.
Finding Color and Shape Patterns in Images.
Technical report, Stanford University, 1999.
Cited in: 2.2.2

Rajdip Dhillon.
Using pause durations to discriminate between lexically ambiguous words and dialog acts in spontaneous speeech.
Journal of the Acoustical Society of America, 123(5):3190–3194, 2008.
Cited in: 4

Danielle Duez.
Second formant locus-nucleus patterns: An investigation of spontaneous French speech.
Speech Communication, 11(4):417–427, 1992.

Cited in: 1.1

Maxine Eskenazi.
Trends in Speaking Styles Research.
In *Third European Conference on Speech Communication and Technology*, 1993.
Cited in: 1.1

Maxine Eskenazi and Anne Lacheret-Dujour.
Exploration of individual strategies in continuous speech.
Speech Communication, 10(3):249–264, 1991.
Cited in: 1.1

Marcus L. Fach.
A Comparison Between Syntactic And Prosodic Phrasing.
In *Proceedings of the European Conference on Speech Communication and Technology*, pages 527–530, Budapest, Hungary, September 1999.
Cited in: 2.1

Cameron S. Fordyce and Mari Ostendorf.
Prosody Prediction for Speech Synthesis using Transformational Rule-based Learning.
In *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia, December 1998.
Cited in: 2.1

Sónia Frota and Vigário Marina.
Tones and Tunes: Typological studies in word and sentence prosody, volume I, chapter Intonational Phrasing in Two Varieties of European Portuguese, pages 265–291.
Mouton de Gruyter, Berlin, 2007.
Cited in: 2

Alfred Charles Gimson.
An introduction to the pronunciation of English.
Edward Arnold London, 4 edition, 1989.
Cited in: 1.1

Frieda Goldman-Eisler.
The distribution of pause durations in speech.
Language and Speech, 4(4):232–237, 1961.
Cited in: 1.2, 4

Björn Granström.
The use of speech synthesis in exploring different speaking styles.
Speech Communication, 11(4):347–355, 1992.
Cited in: 1.1

François Grosjean and Alain Deschamps.

Analyse contrastive des variables temporelles de l'anglais et du français: vitesse de parole et variables composantes, phénomènes d'hésitation.
Phonetica, 31(3–4):144–184, 1975.
Cited in: 1.2

Julia Hirschberg.
Pitch accent in context predicting intonational prominence from text.
Artificial Intelligence, 63(1-2):305–340, October 1993.
Cited in: 2

Julia Hirschberg.
Prosody: Theory and experiment, chapter A corpus-based approach to the study of speaking style, pages 335–350.
Kluwer Academic Publishers, 2000.
Cited in: 1.1

Dirk Hovy, Gopala Krishna Anumanchipalli, Alok Parlikar, Callie Vaughn, Adam Lammert, Ed Hovy, and Alan W Black.
Analysis and Modeling of "Focus" in Context.
In *Proceedings of Interspeech*, Lyon, France, August 2013.
Cited in: 7.1

Andrew J. Hunt and Alan W Black.
Unit selection in a concatenative speech synthesis system using a large speech database.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 373–376, Atlanta, GA, USA, May 1996.
Cited in: 5.2.2

Martin Joos.
Readings in the Sociology of Language, chapter The isolation of styles, pages 181–191. 1968.
Cited in: 1.1

Yeon-Jun Kim and Yung-Hwan Oh.
Prediction of Prosodic Phrase Boundaries Considering Variable Speaking Rate.
In *Proceedings of the 4th International Conference on Spoken Language Processing*, volume 3, pages 1505–1508, Philadelphia, PA, USA, October 1996.
Cited in: 2

Dennis H. Klatt.
The KLATTALK Text-To-Speech Conversion System.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1589–1592, Paris, France, May 1982.
Cited in: 4

Philipp Koehn.
Europarl: A Parallel Corpus for Statistical Machine Translation.

In *Proceedings of Machine Translation Summit*, pages 79–86, Phuket, Thailand, September 2005.
Cited in: 2.4.1, 4.1, 6.1.1

Philipp Koehn, Steven Abney, Julia Hirschberg, and Michael Collins.
Improving Intonational Phrasing With Syntactic Information.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000.
Cited in: 2.1

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst.
Moses: open source toolkit for statistical machine translation.
In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.
Cited in: 5.2.3

Klaus J. Kohler.
Articulatory reduction in different speaking styles.
In *Proceedings of the 13th International Congress of Phonetic Sciences*, volume 12, pages 12–19, 1995.
Cited in: 1.1

John Kominek and Alan W Black.
CMU Arctic Databases for Speech Synthesis.
In *Proceedings of the 5th Speech Synthesis Workshop*, pages 223–224, Pittsburgh, Pennsylvania, June 2004.
Cited in: 2.4.1, 4.1

Diana Krull.
Second formant locus patterns and consonant-vowel coarticulation in spontaneous speech.
Phonetic Experimental Research at the Institute of Linguistics, University of Stockholm, 10:87–108, 1989.
Cited in: 1.1

William Labov.
Phonological Correlates of Social Stratification.
American Anthropologist, 66(6 Part-2):164–176, 1964.
Cited in: 1.1

William Labov.
Sociolinguistic Patterns.
Number 4. University of Pennsylvania Press, 1972.
Cited in: 1.1

Björn Lindblom.
Explaining phonetic variation: A sketch of the H&H theory.
In *Speech production and speech modelling*, pages 403–439. Springer, 1990.
Cited in: 1.1

Fangzhou Liu, Huibin Jia, and Jianhua Tao.
A Maximum Entropy Based Hierarchical Model for Automatic Prosodic Boundary Labeling in Mandarin.
In *Proceedings of the 6th International Symposium on Chinese Spoken Language Processing*, pages 1 –4, Kunming, China, December 2008.
doi: 10.1109/CHINSL.2008.ECP.76.
Cited in: 2.1

Manolis Maragoudakis, Panagiotis Zervas, Nikos Fakotakis, and George Kokkinakis.
A Data-Driven Framework for Intonational Phrase Break Prediction.
In Václav Matoušek and Pavel Mautner, editors, *Text, Speech and Dialogue*, volume 2807 of *Lecture Notes in Computer Science*, pages 189–197. Springer Berlin / Heidelberg, 2003.
ISBN 978-3-540-20024-6.
Cited in: 2.1

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz.
Building a Large Annotated Corpus of English: The Penn Treebank.
Computational Linguistics, 19(2):313–330, 1994.
Cited in: 2.3.2, 2.3.3, 7.1

Erwin Marsi, Martin Reynaert, Antal van den Bosch, Walter Daelemans, and Véronique Hoste.
Learning to predict pitch accents and prosodic boundaries in Dutch.
In *Proceedings of Association for Computational Linguistics*, pages 489–496, July 2003.
URL `http://dx.doi.org/10.3115/1075096.1075158`.
Cited in: 2.1

Mikiko Mashimo, Tomoki Toda, Kiyohiro Shikano, and Wilhelm Nicholas Campbell.
Evaluation of Cross-Language Voice Conversion Based on GMM and Straight.
In *Proceedings of Eurospeech*, pages 361–364, Aalborg, Denmark, September 2001.
Cited in: 2.2.2

Hermann Ney, Ute Essen, and Reinhard Kneser.
On structuring probabilistic dependences in stochastic language modelling.
Computational Linguistics, 8(1):1–38, 1994.
Cited in: 6.1

Nicolas Obin, Pierre Lanchantin, Anne Lacheret, and Xavier Rodet.
Reformulating Prosodic Break Model into Segmental HMMs and Information Fusion.
In *Proceedings of Interspeech*, pages 1829–1832, Florence, Italy, August 2011.
Cited in: 2.1

Franz Josef Och.
Minimum error rate training in statistical machine translation.
In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, 2003.
Cited in: 3.2

Miguel Oliveira.
Pausing Strategies as Means of Information Processing in Spontaneous Narratives.
In *Proceedings of the 1st International Conference on Speech Prosody*, pages 539–542, Aix-en-Provence, France, 2002.
Cited in: 4

Mari Ostendorf, Patti J. Price, and Stefanie Shattuck-Hufnagel.
The Boston University Radio News Corpus.
Technical report, Boston University, March 1995.
URL `http://ssli.ee.washington.edu/papers/radionews-tech.ps`.
Cited in: 2.1, 2.3.1, 2.4.1, 3.3, 4, 4.1, 6.1.1

Sukhada Palkar, Alan W Black, and Alok Parlikar.
Text-to-Speech for Languages without an Orthography.
In *Proceedings of the 24th International conference on Computational Linguistics*, Mumbai, India, December 2012.
Cited in: 6.2.1

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu.
BLEU: a method for automatic evaluation of machine translation.
In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
Cited in: 5.2.3

Alok Parlikar and Alan W Black.
A Grammar Based Approach to Style Specific Phrase Prediction.
In *Proceedings of Interspeech*, pages 2149–2152, Florence, Italy, August 2011.
Cited in: 1.1, 1.5, 2.5

Alok Parlikar and Alan W Black.
Data-Driven Phrasing for Speech Synthesis in Low-Resource Languages.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 2012a.
Cited in: 1.5, 6.3

Alok Parlikar and Alan W Black.
Modeling Pause-Duration for Style-Specific Speech Synthesis.
In *Proceedings of Interspeech*, pages 446–449, Portland, Oregon, USA, September 2012b.
Cited in: 1.5, 4.6

Alok Parlikar and Alan W Black.
Minimum Error Rate Training for Phrasing in Speech Synthesis.
In *Proceedings of 8th Speech Synthesis Workshop*, Barcelona, September 2013.
Cited in: 1.5, 3.6

Alok Parlikar, Alan W Black, and Stephan Vogel.
Improving speech synthesis of machine translation output.
In *Proceedings of Interspeech*, pages 194–197, Makuhari, Japan, September 2010.
Cited in: 1.5, 5.3

F. Pereira and Y. Schabes.
Inside-Outside Reestimation from partially bracket corpora.
In *Proceedings of Association for Computational Linguistics*, pages 128–135, Newark, Delaware, 1992.
Cited in: 2.3, 2.3.3

James R. Phillips.
Online Curve And Surface Fitting.
URL `http://www.zunzun.com`.
Cited in: 3.4

Michael A Picheny, Nathaniel I Durlach, and Louis D Braida.
Speaking clearly for the hard of hearing I: Intelligibility differences between clear and conversational speech.
Journal of Speech, Language, and Hearing Research, 28(1):96, 1985.
Cited in: 1.1

Michael A Picheny, Nathaniel I Durlach, and Louis D Braida.
Speaking clearly for the hard of hearing II: Acoustic characteristics of clear and conversational speech.
Journal of Speech, Language, and Hearing Research, 29(4):434, 1986.
Cited in: 1.1

Kishore Prahallad and Alan W Black.
Segmentation of Monologues in Audio Books for Building Synthetic Voices.
IEEE Transactions on Audio, Speech, and Language Processing, 2010.
Cited in: 2.4.1, 4.1

Kishore Prahallad, Alan W Black, and Ravishankhar Mosur.
Sub-Phonetic Modeling for Capturing Pronunciation Variations for Conversational Speech Synthesis.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 853–856, Toulouse, France, May 2006.
Cited in: 2.3.1, 4.1

Kishore Prahallad, Arthur R. Toth, and Alan W Black.
Automatic Building of Synthetic Voices from Large Multi-Paragraph Speech Databases.

In *Proceedings of Interspeech*, pages 2901–2904, Antwerp, Belgium, August 2007.
Cited in: 4.6

Kishore Prahallad, E. Veera Raghavendra, and Alan W Black.
Learning Speaker-Specific Phrase Breaks for Text-to-Speech Systems.
In *Proceedings of The 7th Speech Synthesis Workshop*, Japan, September 2010.
Cited in: 2, 2.1, 2.2.2, 2.3.1

Kishore Prahallad, E. Naresh Kumar, Venkatesh Keri, S. Rajendran, and Alan W Black.
The IIIT-H Indic Speech Databases.
In *Proceedings of Interspeech*, Portland, OR, USA, September 2012.
Cited in: 6.2.1

Ghislain Putois, Jonathan Chevelu, and Cédric Boidin.
Paraphrase Generation to Improve Text-to-Speech Synthesis.
In *Proceedings of Interspeech*, pages 198–201, Makuhari, Japan, September 2010.
Cited in: 5.2.1

Ian Read and Stephen Cox.
Stochastic and syntactic techniques for predicting phrase breaks.
Computer Speech and Language, 21(3):519–542, 2007.
Cited in: 2.1

P. Roach, G. Knowles, T. Varadi, and S. Arnfield.
MARSEC: A Machine-Readable Spoken English Corpus.
Journal of the International Phonetic Association, 23(1):47–53, 1993.
Cited in: 2.1, 2.4.2

Ken Ross and Mari Ostendorf.
Prediction of abstract prosodic labels for speech synthesis.
Computer Speech and Language, 10(3):155–185, 1996.
Cited in: 2

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas.
A Metric for Distributions with Applications to Image Databases.
International Conference on Computer Vision, 0:59, 1998.
Cited in: 2.2.2

Sittipong Saychum, Chatchawarn Hansakunbuntheung, Nattanun Thatphithakkul, Taneth Ruangrajitpakorn, Chai Wutiwiwatchai, Thepchai Supnithi, Ananlada Chotimongkol, and Ausdang Thangthai.
Categorial-Grammar-Based Phrase Break Prediction.
In *Proceedings of the 8th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 954–957, Khon Kaen, Thailand, May 2011.
Cited in: 2.1

Helmut Schmid and Michaela Atterer.
New Statistical Methods for Phrase Break Prediction.
In *Proceedings of the 20th international conference on Computational Linguistics*, pages 659–665, Geneva, Switzerland, August 2004.
Cited in: 2.1

Marc Schröder and Jürgen Trouvain.
The German text-to-speech synthesis system MARY: A tool for research, development and teaching.
International Journal of Speech Technology, 6(4):365–377, 2003.
Cited in: 4

Kim Silverman, Mary Beckman, John Pitrelli, Mari Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg.
ToBI: A Standard for Labeling English Prosody.
In *Proceedings of 2nd International Conference on Spoken Language Processing*, pages 867–870, Banff, Alberta, Canada, October 1992.
Cited in: 1.2, 2.1

Sunayana Sitaram, Gopala Krishna Anumanchipalli, Justin Chiu, Alok Parlikar, and Alan W Black.
Text to Speech in New Languages without a Standardized Orthography.
In *Proceedings of 8th Speech Synthesis Workshop*, Barcelona, September 2013a.
Cited in: 6.2.1

Sunayana Sitaram, Sukhada Palkar, Yun-Nung Chen, Alok Parlikar, and Alan W Black.
Bootstrapping Text-to-Speech for Speech Processing in Languages without an Orthography.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013b.
Cited in: 6.2.1

Gabriel Skantze, Anna Hjalmarsson, and Catherine Oertel.
Exploring the effects of gaze and pauses in situated human-robot interaction.
In *Proceedings of the 14th annual meeting of the Special Interest Group on Discourse and Dialogue*, 2013.
Cited in: 7.1

Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Shrikanth S. Narayanan.
Factored Translation Models for Enriching Spoken Language Translation with Prosody.
In *Proceedings of Interspeech*, pages 2723–2726, Brisbane, Australia, September 2008.
Cited in: 5.2

Mark Stevenson and Robert Gaizauskas.
Experiments on sentence boundary detection.

In *Proceedings of the sixth conference on Applied natural language processing*, pages 84–89, Morristown, NJ, USA, 2000.
Cited in: 5.2.3

Paul Taylor and Alan W Black.
Assigning phrase breaks from part-of-speech sequences.
Computer Speech and Language, 12:99–117, 1998.
Cited in: 2.1, 2.2.2, 2.3, 3.1, 3.3, 4.3

Laura Mayfield Tomokiyo, Kay Peterson, Alan W Black, and Kevin A. Lenzo.
Intelligibility of Machine Translation Output in Speech Synthesis.
In *Proceedings of Interspeech*, Pittsburgh, September 2006.
Cited in: 5.2

Paul Touati.
Pitch Range and Register in French Political Speech.
In *Proceedings of the 13th International Congress of Phonetic Sciences*, volume 4, pages 244–247, 1995.
Cited in: 1.1

Jean Fox Tree.
Listeners' uses of um and uh in speech comprehension.
Memory and Cognition, 29:320–326, 2001.
Cited in: 5.2.1

Anandaswarup Vadapalli, Peri Bhaskararao, and Kishore Prahallad.
Significance of word-terminal syllables for prediction of phrase breaks in Text to Speech systems for Indian Languages.
In *Proceedings of 8th Speech Synthesis Workshop*, 2013.
Cited in: 6.1.3

C. J. van Rijsbergen.
Information Retrieval.
Butterworth, 1979.
ISBN 0-408-70929-4.
Cited in: 2.2.2, 3.2

Jan P.H. van Santen.
Assignment of segmental duration in text-to-speech synthesis.
Computer Speech and Language, 8(2):95–128, 1994.
Cited in: 2

David J. Wales and Jonathan P. K. Doye.
Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms.
The Journal of Physical Chemistry A, 101:5111–5116, 1997.
Cited in: 3.2

Michelle Q. Wang and Julia Hirschberg.
Automatic Classification of Intonational Phrase Boundaries.
Computer Speech and Language, 6:175–196, 1992.
Cited in: 2.1

Michiko Watanabe, Keikichi Hirose, Yasuharu Den, and Nobuaki Minematsu.
Filled pauses as cues to the complexity of upcoming phrases for native and non-native listeners.
Speech Communication, 50(2):81–94, 2008.
Cited in: 5.2.1

Zhiwei Ying and Xiaohua Shi.
An RNN-based algorithm to detect prosodic phrase for Chinese TTS.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 809–812, Salt Lake City, UT, USA, May 2001.
Cited in: 2.1

Bowen Zhou, Laurent Besacier, and Yuqing Gao.
On Efficient Coupling of ASR and SMT for Speech Translation.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 101–104, Honolulu, HI, USA, April 2007.
Cited in: 5.2

Elena Zvonik.
Pausing and the temporal organization of phrases. An experimental study of read speech.
PhD thesis, University College Dublin, National University of Ireland, November 2004.
Cited in: 4.3

Arnold Zwicky.
On Casual Speech.
In *Eighth Regional Meeting of the Chicago Linguistic Society*, volume 8, pages 607–615, 1972.
Cited in: 1.1