Text-to-Speech Synthesis Using Found Data for Low-Resource Languages

Erica Cooper

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2019

ABSTRACT

Text-to-Speech Synthesis Using Found Data for Low-Resource Languages

Erica Cooper

Text-to-speech synthesis is a key component of interactive, speech-based systems. Typically, building a high-quality voice requires collecting dozens of hours of speech from a single professional speaker in an anechoic chamber with a high-quality microphone. There are about 7,000 languages spoken in the world, and most do not enjoy the speech research attention historically paid to such languages as English, Spanish, Mandarin, and Japanese. Speakers of these so-called "low-resource languages" therefore do not equally benefit from these technological advances. While it takes a great deal of time and resources to collect a traditional text-to-speech corpus for a given language, we may instead be able to make use of various sources of "found" data which may be available. In particular, sources such as radio broadcast news and ASR corpora are available for many languages. While this kind of data does not exactly match what one would collect for a more standard TTS corpus, it may nevertheless contain parts which are usable for producing natural and intelligible parametric TTS voices.

In the first part of this thesis, we examine various types of found speech data in comparison with data collected for TTS, in terms of a variety of acoustic and prosodic features. We find that radio broadcast news in particular is a good match. Audiobooks may also be a good match despite their largely more expressive style, and certain speakers in conversational and read ASR corpora also resemble TTS speakers in their manner of speaking and thus their data may be usable for training TTS voices.

In the rest of the thesis, we conduct a variety of experiments in training voices on non-traditional sources of data, such as ASR data, radio broadcast news, and audiobooks. We aim to discover which methods produce the most intelligible and natural-sounding voices, focusing on three main approaches:

- **Training data subset selection.** In noisy, heterogeneous data sources, we may wish to locate subsets of the data that are well-suited for building voices, based on acoustic and prosodic features that are known to correspond with TTS-style speech, while excluding utterances that introduce noise or other artifacts. We find that choosing subsets of speakers for training data can result in voices that are more intelligible.

- **Augmenting the frontend feature set with new features.** In cleaner sources of found data, we may wish to train voices on all of the data, but we may get improvements in naturalness by including acoustic and prosodic features at the frontend and synthesizing in a manner that better matches the TTS style. We find that this approach is promising for creating more natural-sounding voices, regardless of the underlying acoustic model.

- **Adaptation.** Another way to make use of high-quality data while also including informative acoustic and prosodic features is to adapt to subsets, rather than to select and train only on subsets. We also experiment with training on mixed high- and low-quality data, and adapting towards the high-quality set, which produces more intelligible voices than training on either type of data by itself.

We hope that our findings may serve as guidelines for anyone wishing to build their own TTS voice using non-traditional sources of found data.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This thesis would not have been possible without the mentorship and guidance of my advisor, Julia Hirschberg. She taught me how to conduct research, and her support of this project has meant a great deal. I will always be tremendously grateful.

I would also like to thank my dissertation committee – Kathleen McKeown, Mike Collins, Richard Sproat, and Alan Black. Their guidance and advice over the years have been invaluable.

I would also like to express my gratitude to the Google Speech Team for providing me with numerous opportunities to learn and expand my skills during my time as an intern. I would like to thank Martin Jansche for his mentorship over the years, for encouraging me to apply to grad school, and for introducing me to Julia. Terry Tai taught me a great deal about the software engineering aspects of large-scale speech systems, and I am a better coder because of him. Heiga Zen taught me most of what I know about parametric synthesis, and has continued to provide helpful advice on my thesis work over the years. I was also very fortunate to work with Yannis Agiomyrgiannakis, Alexander Gutkin, Richard Sproat, Monika Podsiadło, Xavi Gonzalvo, Phil Gross, Michiel Bacchiani, and Michael Riley.

I would also like to thank all those who have helped with this project. Andrew Rosenberg and Raul Fernandez have provided us with industry-grade TTS data, as well as assistance in automatically evaluating our voices. Alan Black provided scripts and assistance with using Festival for different languages. Meredith Brown shared knowledge for getting started with our crowdsourcing work on Mechanical Turk. This project also would not have been possible without the help of our undergraduate and masters project students over the years, who prepared data, ran experiments, built crucial project infrastructure, and contributed language knowledge: Alison Chang, Yocheved Levitan, Luise Valentin Rygaard, Olivia Lundelius, Cindy Wang, Mert Usslaki, David Tofu, Emily Li, Kai-Zhan Lee, Elshadai Tesfaye Biru, and Yishak Tofik Mohammed. I am also very grateful to the Columbia CRF and CUIT staff for their ongoing technical support of the systems on which we

run experiments.

I am grateful to the many friends and colleagues over the years whose camaraderie and enlightening discussions have helped me grow as a researcher and have made my PhD years a true joy. I would like to thank my past and present labmates at the Columbia Speech Lab: Victor Soto, Sarah Ita Levitan, Rose Sloan, Morgan Ulinski, Nishi Cestero, Xi Chen, Brenda Yang, Daniel Bauer, Fadi Biadsy, Bob Coyne, Rivka Levitan, Gideon Mendels, Laura Willson, Anna Prokofieva, Michelle Levine, Svetlana Stoyanchev, and Shirley Xia. I would also like to thank my friends in the Computer Science department: Avner May, Arthi Ramachandran, Melanie Kambadur, Bingyi Cao, Heba Elfardy, Eva Sitaridi, and Kapil Thadani. Also, thanks to my friends and colleagues in the speech and TTS community at large: Bhuvana Ramabhadran, Sandrine Brognaux, Rasmus Dall, Gustav Eje Henter, Sebastien Le Maguer, Avashna Govender, Joe Mendelson, Paweł Cyrta, Srikanth Ronanki, and Oliver Watts.

Finally, I would like to thank my partner and best friend, Leonard Witzel, for his patient support, kindness, and companionship over the years, and for his creativity, sense of humor, and unique perspective. Without him, these years would have been much more difficult.

Chapter 1

Introduction

Recent advances in speech technology have led to a proliferation of speech-enabled consumer applications. From virtual assistants such as Apple's Siri and Amazon's Echo to Panasonic's translating airport megaphone and Samsung's talking refrigerator, it can seem like speech technology is everywhere. However, this is only truly the case for languages which have been fortunate enough to receive the corporate or government resources and research attention required to collect and annotate the large amounts of data and linguistic information needed to build precise, domain-appropriate speech models. Text-to-speech synthesis (TTS) is a key component of interactive, speech-based technology such as spoken dialogue systems, virtual personal assistants and speech-to-speech translation, and typically, building a high-quality voice requires collecting dozens of hours of speech recorded from a single professional speaker in an anechoic chamber with a high-quality microphone. There are approximately 7,000 languages in the world, and most do not enjoy the speech research attention historically paid to such languages as English, Spanish, Mandarin, and Japanese; speakers of these so-called "low-resource languages" therefore do not equally benefit from these technological advances. Thus, they lack access to technologies they can use to communicate and search for information in their own language by voice – a major accessibility issue for those who lack the ability to read. Furthermore, in this global age, access to language technology such as speech translation becomes important not only for travelers but for medical professionals, emergency response staff, corporations, journalists, the military and law enforcement.

Researchers have previously explored many different techniques for building voices for languages where large amounts of speech data are not available. Approaches such as bootstrapping resources across similar languages, creating automatic (supervised and unsupervised) labeling approaches, and collecting new data have all proven useful. For instance, (Dijkstra *et al.*, 2004) created a synthesizer for Frisian, a minority language of the Netherlands, by bootstrapping data and resources from Dutch and Spanish. Similarly, (Yang *et al.*, 2015) built TTS for Tibetan by extending a Mandarin synthesizer. (Ekpenyong *et al.*, 2014) built a synthesizer for Ibibio, a Nigerian tone language, by recording a small amount of professional speech and porting an existing frontend for Spanish. Moreover, (Sitaram *et al.*, 2013) bootstrapped synthetic voices for languages without a standardized orthography, by performing cross-lingual phonetic decoding on untranscribed audio. Language adaptation is another form of bootstrapping from higher-resource languages when one only has minimal target language data: (Anumanchipalli and Black, 2010) pooled data from nine different source languages and performed a phoneme mapping to create adapted voices for German and Telugu using as few as 20 utterances in the target languages. There have also been considerable efforts to build language-independent tools to streamline the process of building voices for new languages. Festvox (Black and Lenzo, 2003) is a framework that provides general tools for building voices in any language. It has enabled the creation of unit-selection voices in many languages. For instance, (Mariam *et al.*, 2004) used Festvox to build a voice for Amharic using their own recorded data, and identified the language-specific challenges of Amharic regarding epenthesis and syllabification. (Kominek *et al.*, 2007) expanded upon Festvox by creating online tools for individuals to easily record and label speech, and tested them in a classroom setting on a variety of languages. Users record their own audio from texts automatically selected for maximal grapheme coverage, and incrementally build a lexicon starting with the most common words to create baseline rules, and then manually accepting or correcting pronunciations for additional words. Similarly, methods and tools for building TTS frontend modules in the absence of large amounts of hand-annotated data were developed in (Watts *et al.*, 2013) for the purpose of building voices in arbitrary new languages, but without assuming knowledge of the language. They used a lightly-supervised active learning based data selection approach to choose neutral utterances from audiobooks, and created a vector space model by computing co-occurrence statistics about the distribution of graphemes in text, which were used as linguistic units instead of phonemes. More recently, (Gutkin *et al.*, 2016) collected their own data to build a TTS database for Bangla by using “multiple ordinary speakers” instead of one professional speaker, but choosing speakers with similar vocal characteristics to each other.

While it takes a great deal of time and resources to collect a traditional text-to-speech corpus for a given language, we may instead be able to make use of various sources of "found" data which may be available. In particular, sources such as radio broadcast news, audiobooks, and podcasts are available online in many languages. While this kind of data does not exactly match what one would collect for a more standard TTS corpus, it may nevertheless contain a substantial amount of speech from each speaker, the speakers may be professionals, and the recording conditions may be high-quality. The major difference is that TTS speakers are typically instructed to speak as consistently as possible, without varying their voice quality, speaking style, pitch, volume, or tempo significantly (Matoušek *et al.*, 2008), whereas even broadcast news anchors will have some variance in their speech, even when they are speaking in an otherwise predominantly neutral style. Furthermore, for many years, TTS research has been dominated by the *unit selection* paradigm – the simple concatenation of words and phrases from very large datasets recorded by a professional speaker in a sound booth, and laboriously hand-annotated for any phenomenon whose variation is desired. Unfortunately, this paradigm typically results in degraded quality of synthetic speech when producing of out-of-vocabulary words (e.g. proper names) or prosodic variations which have not been explicitly recorded and annotated. This makes the creation of unit-selection TTS systems a lengthy, laborious, and expensive process, typically only feasible for companies with large budgets. In fact, it can cost major companies over $1 million to create a new TTS voice from beginning to end. This has been a major factor in limiting the creation of TTS systems to those of immediate commercial value. However, the more recent emergence of *parametric* approaches to synthesis, often implemented using Hidden Markov Models (HMMs) or neural network based architectures, now makes it possible to create TTS systems quickly and inexpensively in a research environment. Such systems can be trained on much smaller amounts of speech from multiple speakers recorded in less controlled environments and for purposes other than synthesis.

In this work, we investigate the following research questions:

- Can we quantify the salient acoustic and prosodic similarities and differences between traditional TTS corpora and other types of data?

- Can we build and evaluate TTS voices from different sources of found data that are not typically used for TTS?

- Can we identify subsets of a non-traditional source of data for TTS, such as radio broadcast news or ASR corpora, based on acoustic and prosodic features, to select parts of those corpora that are the most similar to the kind of data found in a traditional TTS corpus, for low-resource languages for which a TTS corpus is not available?
- Can we identify voice modeling approaches that are best suited to this type of data, making use of knowledge about salient acoustic and prosodic features?
- Can these approaches be used to create voices from found data that are reasonably natural-sounding and intelligible?

In this thesis, we present experiments in creating voices for US English as well as in low-resource languages such as Amharic and Turkish using radio broadcast news, ASR telephone corpora, and audiobooks using different approaches such as subset selection and adaptive training, with crowd-sourced evaluations for naturalness and intelligibility.

Chapter 2

Tools and Evaluation

2.1 Introduction

The advent of statistical parametric speech synthesis (SPSS) such as Hidden Markov Model (HMM) based synthesis and neural network based synthesis has enabled the creation of voices without necessarily requiring large amounts of high-quality, single-speaker data. We use a variety of open-source tools for frontend processing and for training parametric voice models using both HMMs and neural networks. Parametric synthesis is an approach that learns voice model parameters from data. New speech is synthesized by using the trained models to generate the appropriate acoustic feature sequence for the given text, and then that sequence is converted into an audio waveform using a vocoder. This approach is in contrast with the *unit selection* approach to TTS, in which large amounts of recorded speech from a voice talent are segmented and rearranged to form new utterances. While the process of using a vocoder to generate completely new audio from generated acoustic features typically results in audio that has a "buzzy" sound that is less natural than using pieces of actual recorded audio, parametric synthesis nevertheless has a number of advantages over unit selection: first, a parametric "voice" consists only of the model parameters of the trained voice, rather than the entire recorded training corpus, resulting in a smaller footprint. Second, unit selection absolutely requires that the entire corpus be spoken by the same speaker, in the same style, under the same recording conditions – any variation will result in an audible change where the units are joined together. Parametric synthesis, on the other hand, learns *average* statistics about what speech sounds like – thus, we can train voices on data that contains mixed speakers, recording conditions, and styles. This makes parametric synthesis especially well-suited for our task of training voices on data which was collected for other purposes. Third, model parameters can be *adapted* to different speakers or styles, allowing for very flexible creation of different types of voices.

2.2 Hidden Markov Model Based Synthesis

For parametric voice training and synthesis, we primarily work within the framework of HTS (H Triple-S, The HMM-based Speech Synthesis System) (Zen *et al.*, 2007), an open-source parametric HMM-based synthesis system. HTS is a modified version of HTK, the Hidden Markov Model Toolkit (Young *et al.*, 1995). HTS implements all of the basic algorithms needed to train acoustic models for an HMM-based synthesizer and provides basic recipes for speaker-independent (SI) and speaker-adaptive training (SAT), as well as allowing for very fine-grained control over model training and adaptation.

An overview of HMM-based synthesis as implemented in HTS is shown in Figure 2.1. Training data is provided as aligned linguistic features (extracted by a separate frontend) and acoustic features (extracted from the audio signal). Linguistic features are represented as phonemes in context, where context can include arbitrary features chosen by the user. The standard set of contextual features includes previous two and next two phonemes, syllable-related features such as the vowel of the current syllable, stress information, and word- and phrase-level features such as the position of the current word or phrase in the utterance. Standard acoustic features consist of log f0 and spectral features represented as mel-generalized cepstral coefficients (MGC), as well as deltas and delta-deltas of those values, each modeled as their own stream.

Each phoneme-in-context gets modeled as an HMM with five states to allow for change in the sound of that phoneme over its beginning, middle, and end. Due to the large set of contextual features, sparsity is an issue – many combinations of phonemes and context will only be seen in the data a small number of times, and most will never be seen at all. To combat this, a decision tree is built during training where yes/no questions are asked about the phonemes and their context (specified in a user-provided, language-dependent questions file), and similar models are clustered together – their data is pooled, and their parameters are learned together and shared. At synthesis

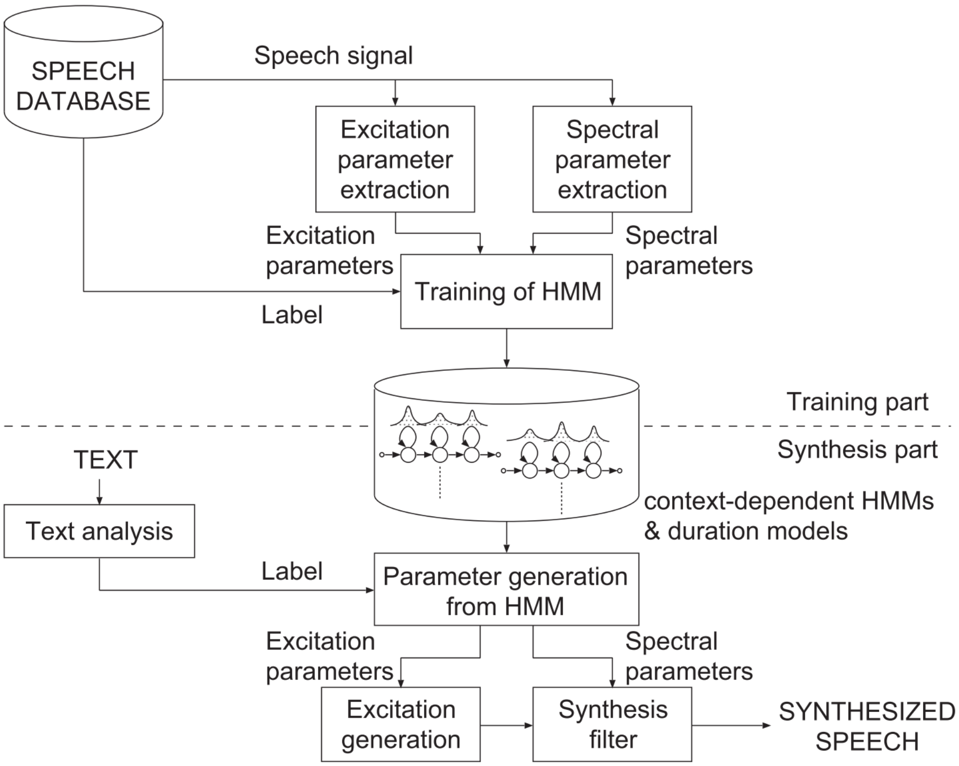


Figure 2.1: HMM-based speech synthesis. (Black *et al.*, 2007)

time, the text is converted into a contextual-feature representation by the frontend, the appropriate models are selected by traversing the decision trees, and a sequence of acoustic parameters are generated by those models, which are lastly converted into audio by the vocoder.

The HTS speaker-independent (SI) recipe starts by initializing monophone models and then does five rounds of embedded re-estimation, which uses the Baum-Welch algorithm to get a maximum-likelihood estimate of the model parameters. Then, the monophone models are copied into full-context ones, and embedded re-estimation is repeated for the full-context models. Next, the decision tree is constructed and clustering is performed, followed by clustered embedded re-estimation. Then there is another round of re-clustering with the re-estimated models, and lastly, a final re-estimation step.

The speaker-adaptive training (SAT) recipe first trains a base model in the same manner as the SI recipe. Then, three rounds of speaker-adaptive training are performed in which first, the CMLLR (constrained maximum-likelihood linear regression) transforms for each speaker are estimated and then the AVM (average voice model) HMM and duration models are re-estimated. Finally, the target speaker's transform is applied to the AVM and speech is synthesized for the target speaker.

2.3 Neural Network Based Synthesis

More recently, the availability of greater computational resources and more data has enabled deep learning architectures to be leveraged for creating high-quality parametric text-to-speech models (Kang *et al.*, 2013; Zen *et al.*, 2013). Neural network based synthesis replaces the decision trees and HMMs of HMM-based synthesis with a deep neural network as the regression model. This approach has recently produced very high-quality voices, and addresses some of the naturalness issues common to HMM-based voices. (Merritt *et al.*, 2015) found that the across-class averaging resulting from decision tree-based context clustering is a major detractor of naturalness in HMM voice quality, and (Watts *et al.*, 2016) found that replacing the decision trees with deep neural networks (DNNs) and the production of frame-level rather than state-level predictions substantially improved naturalness. Furthermore, (Wang *et al.*, 2016) found that the modeling approach selected can vastly reduce the amount of training data required to produce voices of comparable quality – an HMM system trained on 100 hours of data was comparable in f0 correlation (an objective measure of naturalness) to a DNN system using only 10 hours, and a DNN system trained on 100 hours of data was comparable to a DBLSTM-RNN (deep bidirectional long short-term memory recurrent neural network) system trained on only 10 hours. While these results were for voices trained on single-speaker data collected specifically for TTS, we may also benefit from using modeling approaches that can produce higher-quality voices with less data.

The Merlin toolkit for neural network based speech synthesis (Wu *et al.*, 2016) is a modular, extensible toolkit for training synthetic voices. Merlin provides Kaldi-style (Povey *et al.*, 2011) recipes – Kaldi is a popular open-source research toolkit for automatic speech recognition, and one of the main goals of the Kaldi project was to provide complete recipes for building speech recognition systems that work on commonly-used corpora such as those available from the Linguistic Data Consortium. Merlin allows the user to configure the number and type of layers in the neural network voice model, and also allows for different choices of vocoder. While Merlin accepts HTS-style full-context labels, it further processes those input features to be suitable for a neural network. Neural networks require numeric features whereas the decision tree regression of HTS requires categorical features, so the categories (e.g. phoneme identities) are represented as a series of one-hot encodings. Then, the linguistic phoneme-by-phoneme representation is converted into a temporal representation, in which phonemes are broken down into states and states into frames, with positional information of the frame within the state being added to the input feature representation. A separate durational model must be trained first, followed by acoustic model training.

The default Merlin speaker-independent "build your own voice" recipe uses WORLD (Morise *et al.*, 2016) for acoustic feature extraction and vocoding, and trains models consisting of 6 tanh (hyperbolic tangent) layers each of size 1024, with a linear activation function at the output layer. Batch size is 64 for the duration model and 256 for the acoustic model. Learning rate is fixed at 0.002, momentum at 0.3, and number of training epochs at 25.

The Merlin speaker adaptation recipe implements two different types of adaptation (described in (Bollepalli *et al.*, 2017)): back-propagating the adaptation data through the model to re-tune all the weights ('fine-tune'); and "Learn Hidden Unit Contributions" (LHUC), which recombines hidden units based on the adaptation data (Swietojanski *et al.*, 2016).

2.4 Frontend Processing

For frontend text processing, we use the Festival Speech Synthesis System (Black *et al.*, 2014), which has a very modular architecture allowing for providing custom pronunciation lexica, tokenization, post-lexical rules, etc., as well as phonetic forced alignment with audio data using the Ergodic Hidden Markov Model (EHMM) algorithm (Prahallad *et al.*, 2006), and which integrates well with HTS and Merlin by generating appropriately-formatted label files from text. Although Festival is capable of operating as a complete text-to-speech system, we mainly use its frontend processing capabilities. Festival generates structured Utterance files from text, which contain hierarchical information about words, phrases, stress, and pronunciation, as well as alignment information in the case of matched audio training data. These files can be flattened into the HTS-style phones-in-context label format for use with both HTS and Merlin. The standard set of contextual features for the phoneme-level flattened labels include previous two, current, and next two phonemes; the position of the current phoneme in the syllable; the position of the current syllable in the word; whether the syllable is stressed or not; position of the current word in the phrase; etc. The full set of standard contextual features from the HTS documentation is inluded for reference in Appendix A.

2.5 Evaluation

We explored the use of crowdsourcing to evaluate the voices we created, as well as a number of methods of objective evaluation. While subjective measures are preferable for an application such as text-to-speech synthesis, where it is most important that human listeners can understand the speech and want to listen to it, the nature of the low-resource setting necessitates the use of objective measures as well, to facilitate experimental turnaround time.

2.5.1 Crowdsourced Evaluation

To evaluate the naturalness and intelligibility of each voice we produced, we published crowdsourced listening tests online, using Amazon Mechanical Turk (MTurk), a popular crowdsourcing platform. To restrict our listeners to native English speakers, we required workers to complete a qualification test before completing any of our tasks, in which we asked which languages they have spoken since birth, from a list of languages. For evaluations in a given language, we only allowed workers who selected that language and no more than two other languages, in order to exclude those who might select, e.g., all of the languages in an attempt to cheat the system. We also restricted our tasks' visibility to workers within the United States.

2.5.1.1 Transcription Task for Intelligibility

We evaluated the intelligibility of the voices we created using standard TTS evaluation metrics for intelligibility (Buchholz *et al.*, 2013). Initially, we produced 10 syntactically-sound but semantically unpredictable sentences (SUS) of the form `det adj noun verb det adj noun`, which is a standard form used for TTS evaluation such as in the Blizzard challenge (Black and Tokuda, 2005) (for example, "the operational waves scattered a doubtful account."). We synthesized this set of sentences with each of our trained voices that we wished to evaluate; we also included one semantically-predictable sentence spoken clearly as an attention check question. Workers were asked to transcribe what they heard for each of the eleven sentences, presented in random order. Since the sentences were the same for each voice, to enable a sensible comparison across voices, workers were only allowed to transcribe sentences for one voice, to remove any bias arising from the workers hearing repeat sentences and remembering them. Five workers transcribed each sentence for each voice.

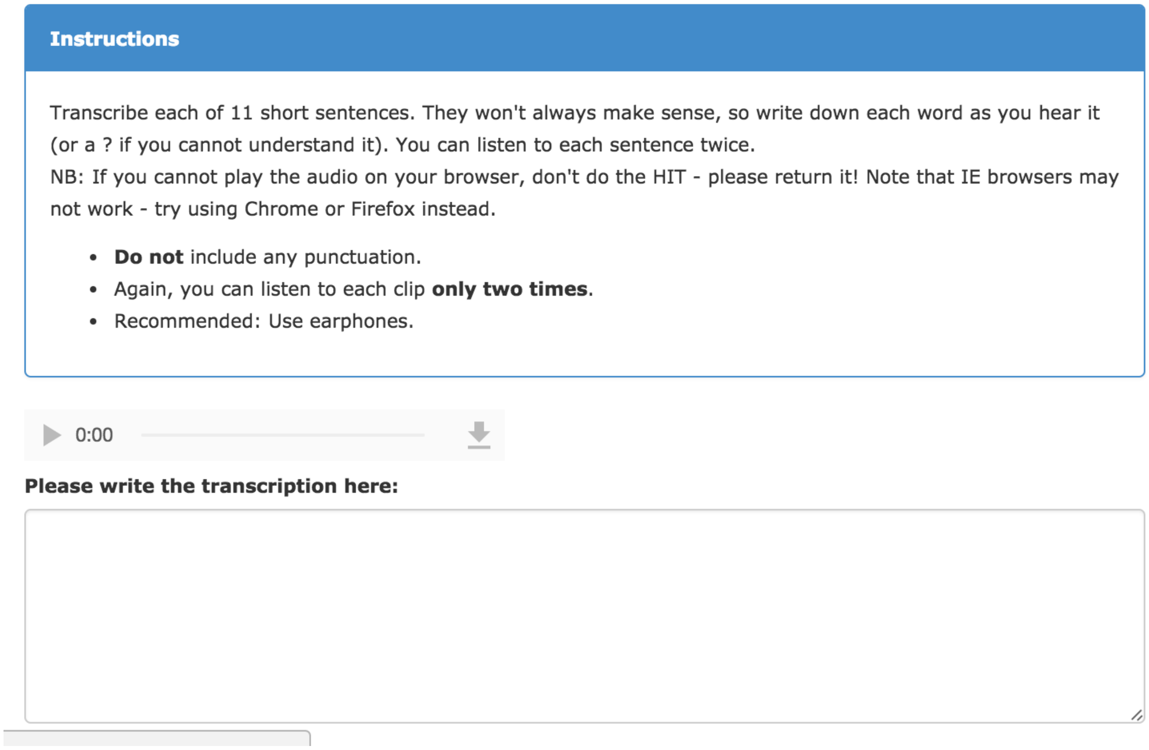
After considering that listener variability may be large and we should attempt to control for it, we later switched to a Latin-square configuration for the intelligibility tests. This ensures not only that no listener hears the same sentence twice, but also that all listeners hear utterances from all voices, thus ensuring that listener variability gets distributed equally across voices.

After sentences were transcribed, we computed word error rate (WER) for each voice (averaging over each of the five workers) to measure intelligibility, comparing the transcriptions to the text that was actually synthesized. Word error rate is computed as follows:

$$WER = \frac{I + D + S}{N}$$

Where I is the number of insertions, D is the number of deletions, S is the number of substitutions, and N is the total number of words in the ground truth sentence. Note that since listeners do not know how many words are in the truth transcription, and there can potentially be many insertions, it is possible to obtain word error rates greater than 100%.

Since the transcriptions were typed by humans, they were prone to typographical errors and misspellings, which we hand-corrected. We also allowed singular/plural confusions, such as “musical” / “musicals,” but we did not allow confusions between words with the same stem, such as “fragrant” / “fragrance.” We also allowed compound word variants, such as “blackbird” / “black bird.”



Instructions

Transcribe each of 11 short sentences. They won't always make sense, so write down each word as you hear it (or a ? if you cannot understand it). You can listen to each sentence twice.
NB: If you cannot play the audio on your browser, don't do the HIT - please return it! Note that IE browsers may not work - try using Chrome or Firefox instead.

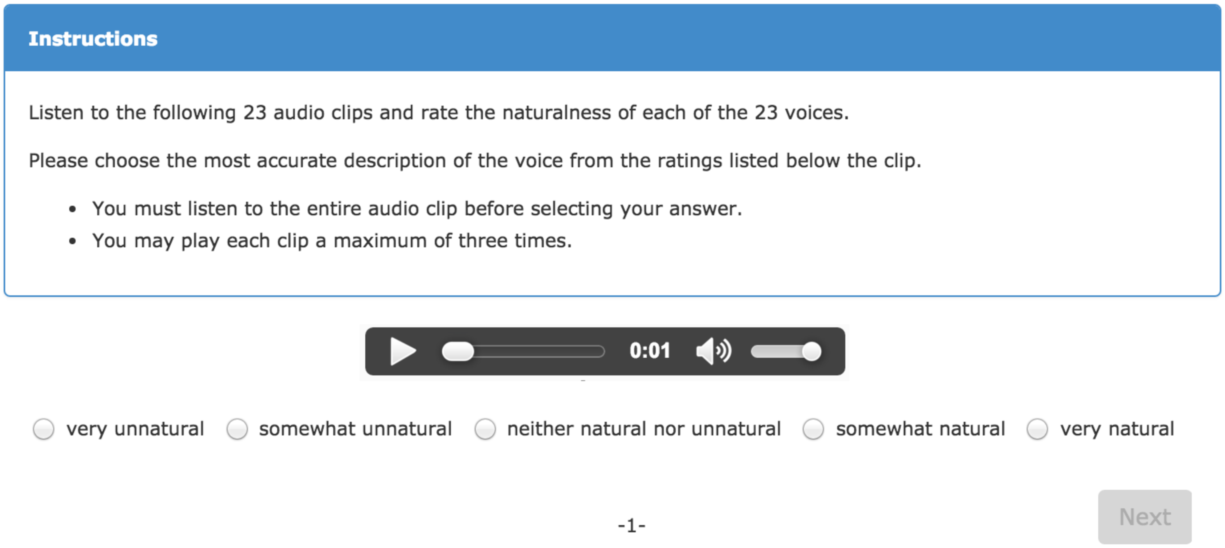
- **Do not** include any punctuation.
- Again, you can listen to each clip **only two times**.
- Recommended: Use earphones.

Please write the transcription here:

Figure 2.2: Pairwise preference task on Amazon Mechanical Turk

2.5.1.2 Mean Opinion Score Test for Naturalness

Our first experiment was modeled after the Mean Opinion Score (MOS) listening test outlined in (Georgila *et al.*, 2012) in which workers were asked to evaluate the naturalness of 12 spoken utterances by selecting from a 5-point Likert scale, from 1 = very unnatural, 2 = somewhat unnatural, 3 = neither natural nor unnatural, 4 = somewhat natural, and 5 = very natural. We chose lexically neutral sentences of varying length from the fable “Jack and the Beanstalk” and synthesized them with our baseline and test voices. In addition to utterances synthesized with our test voices (initially, 20 voices) and the baseline voice, we also recorded a human voice saying the same sentences, and also generated a set of utterances using a very robotic-sounding voice, generated using the "Zarvox" voice Mac OS X's `say` command. The human and robotic voices were used as references to determine whether to accept the submitted work. If they received unreasonable ratings (anything but 4 or 5 for the human voice, or 1 or 2 for the robotic voice) the submission was not used. We randomized the order of the test audio files but ensured that the reference voices were always in the last half of the playlist to avoid skewing listeners' opinions. In addition, workers were not allowed to rate a voice until they had listened to the entire audio file being rated. Each task consisted of the same sentence spoken by each of the 23 voices. Every task was completed by 5 unique individuals.



Instructions

Listen to the following 23 audio clips and rate the naturalness of each of the 23 voices.

Please choose the most accurate description of the voice from the ratings listed below the clip.

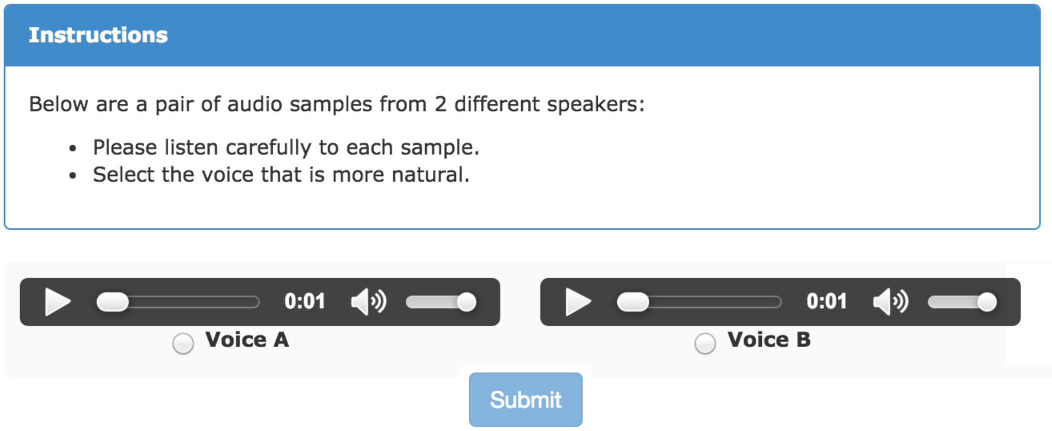
- You must listen to the entire audio clip before selecting your answer.
- You may play each clip a maximum of three times.

○ very unnatural ○ somewhat unnatural ○ neither natural nor unnatural ○ somewhat natural ○ very natural

Figure 2.3: Mean Opinion Score task on Amazon Mechanical Turk

2.5.1.3 Pairwise Preference Test for Naturalness

Since the MOS test required that each rater listen to the same sentence spoken by many different voices, and since we failed to find substantial differences between the ratings of the test voices in our preliminary experiments, we suspected that a MOS test in which raters were asked to compare 23 voices might have been overwhelming for the listeners, in effect precluding meaningful comparisons. We therefore designed a second task, a standard pairwise comparison (Buchholz *et* *al.*, 2013) between the baseline voice and each test voice. Each task thus contained only two audio files, the same sentence spoken by the baseline voice and one of our test voices. Workers could rate as many or as few pairs of utterances as they wished. Half of the sentences were presented in A/B order and the other half in B/A order, to avoid possible order effects. We ensured that raters listened to both audio files entirely before they were allowed to submit their preference. Raters were given a forced choice, i.e. there was not a "no preference" option. We used the same 12 sentences as in our MOS test and collected 5 ratings from unique individuals per baseline/test-voice pair, thereby obtaining 60 ratings per voice. We compute significance of the preferences using a z-test for a proportion as in (Buchholz *et al.*, 2013). We do not define "naturalness" for listeners, since we mainly want to get their opinions and impressions without biasing them to focus on any particular aspects of the voices.



Instructions

Below are a pair of audio samples from 2 different speakers:

- Please listen carefully to each sample.
- Select the voice that is more natural.

Figure 2.4: Pairwise preference task on Amazon Mechanical Turk

2.5.2 Automatic Evaluation

A limitation of our experimental approach is the long turnaround time for crowdsourcing voice transcriptions. Since each worker is allowed to transcribe only a single voice, evaluation proceeds slowly regardless of individual workers' interest in the task. Evaluation proceeded even more slowly once we started evaluating voices for low-resource languages. (Pavlick *et al.*, 2014) has revealed a linguistically-diverse community of workers on Mechanical Turk – a sample of about 3,200 bilingual workers revealed native fluency in over 80 different languages, 35 of which had more than 20 speakers, and 9 of which are languages for which we have both read and conversational ASR data from the BABEL program (Harper, 2011). However, Amazon has since placed more restrictions on workers outside of the United States, which, along with the restrictions of our task, has made it very difficult to use crowdsourcing to evaluate voices for low-resource languages. These limitations have led us to investigate the possibility of using Automatic Speech Recognition (ASR) to evaluate intelligibility. Although an ASR system will not interpret a voice exactly as a human would, depending heavily upon the type of data on which it was trained, we thought it worthwhile to see whether it might be a feasible evaluation tool that correlates with human performance.

We tested three different general-purpose, state-of-the-art, industry-level ASR APIs (Application Programming Interfaces) to determine the viability of their use for evaluating voices: wit.ai[1], a natural language API toolkit owned by Facebook that includes ASR; Watson[2], IBM's API for cognitive applications; and the Google Cloud Speech API limited preview[3]. We decided to try APIs rather than building our own ASR for pilot work, and also because using state-of-the-art recognizers would hopefully provide the best possible proxy for a human listener. Our hypothesis is that some of these recognizers may correlate well with human listeners, and we can use this as a first step to pick only our best candidate voices to send to Mechanical Turk for further human evaluation. We found strong correlations with human intelligibility judgments for synthesized voices in the experiments described in Chapter 4, Section 4.4.2.

We also use more traditional objective measures to evaluate voices. Objective measures for intelligibility are typically used for measuring signal loss of natural speech in noisy environments or over a noisy transmission line and are often only applicable in limited circumstances (Schmidt-Nielsen, 1992); it is rare for these measures to be used for evaluating synthetic speech. However, (Valentini-Botinhao *et al.*, 2011) explored the use of a variety of such metrics to evaluate speech from a state-of-the-art HMM synthesizer under a number of additive noise conditions, finding that some measures correlated well with human intelligibility ratings. While a voice trained on high-quality, single-speaker, TTS-specific data and played in a noisy environment is likely to have different intelligibility issues than a voice originally trained on noisy data, it may nevertheless

[1] `https://wit.ai/`; accessed March 1, 2017.

[2] `https://www.ibm.com/watson/developercloud/speech-to-text.html`; accessed March 1, 2017.

[3] `https://cloud.google.com/speech/`, accessed March 1, 2017.

be worthwhile to see whether these measures correlate with human judgments for our voices as well. In particular, mean mel-cepstral distortion (MCD) (Kubichek, 1993) is a popular measure of overall voice quality that has been shown to correlate highly with Diagnostic Acceptability Measure (DAM) (Voiers, 1977), a general measure of speech acceptability that incorporates intelligibility and pleasantness, which was also developed for evaluating voice transmission systems. While MCD requires resynthesis of utterances from a held-out test set and does not measure distortions in the pitch contour, it may nevertheless be another useful metric to incorporate into our evaluation, and we explore correlations with human judgments in Chapter 4, Section 4.5.3.

Chapter 3

Characteristics of TTS Data vs. Other Genres

3.1 Introduction

Collecting the type of data required to build a high-quality TTS voice is typically very costly. Anecdotally it costs around one million dollars to create a completely new production-quality voice, and thus it is only undertaken with a major economic motivation. Typically, a professional voice talent reads dozens of hours of text with good lexical coverage of the target domain in a soundproof room with a high-quality microphone and in as neutral and even a style as possible. They are typically instructed to maintain constant f0, energy, speaking rate, and articulation throughout (Matoušek *et al.*, 2008). Without the resources to collect such data, is it still possible to create a high-quality voice? With the advent of statistical parametric speech synthesis (SPSS), it is possible to create voices without necessarily having to collect large amounts of high-quality, single-speaker, in-domain speech. Furthermore, large amounts of available speech such as audiobooks and radio broadcast news present a promising source of data for building new voices. In this chapter, we examine a number of corpora in different genres, collected for different purposes, in order to compare their similarities and differences with respect to various acoustic and prosodic features. We aim to determine whether TTS corpora do in fact follow the "standard" TTS speaking style, whether other forms of professional and non-professional speech differ substantially from the TTS style, and which features are more salient in differentiating the speech genres. Preliminary work for this chapter has been published in (Cooper *et al.*, 2018).

3.2 Related Work

TTS speakers are typically instructed to speak as consistently as possible, without varying their voice quality, speaking style, pitch, volume, or tempo significantly (Matoušek *et al.*, 2008). This is different from other forms of professional speech in that even with the relatively neutral content of broadcast news, anchors will still have some variance in their speech. Audiobooks present an even greater challenge, with an intentionally more expressive reading style and different character voices. Nevertheless, (Chalamandaris *et al.*, 2014; Stan *et al.*, 2013; Braunschweiler and Buchholz, 2011) have had success in building voices from audiobook data by identifying and using the most neutral and highest-quality utterances. Furthermore, in our own prior work (Cooper *et al.*, 2016b; Cooper *et al.*, 2016a; Cooper *et al.*, 2017), we have created more natural-sounding voices out of radio broadcast news speech and data collected for automatic speech recognition (ASR) by selecting training utterances based on acoustic and prosodic criteria motivated by knowledge of what makes a "good" TTS voice. In this chapter we will validate these assumptions about TTS data empirically and identify similarities and differences when we compare them to other genres. We do this for the purpose not only of identifying which genres may be most suitable for building TTS voices, but also determining which utterances or speakers within those genres should be selected or discarded in the process as well.

3.3 Corpora

We examine the similarities and differences in various acoustic and prosodic features in a number of different corpora. Such corpora include TTS recordings, audiobook speech, radio broadcast news, and telephone conversations recorded to train ASR systems in a variety of languages, in both conversational and read scripted styles.

3.3.1 TTS Corpora

3.3.1.1 Research TTS Corpora

The CMU ARCTIC databases (Kominek and Black, 2003) were collected in studio conditions for unit selection synthesis research and consist of approximately one hour per speaker of phonetically-balanced sentences collected from out-of-copyright texts. Currently, the database consists of two male and two female US English speakers, as well as Canadian, Scottish, and Indian English male speakers. We examine the US English data in this work.

The SWARA corpus (Stan *et al.*, 2017) contains studio-quality recordings from 17 volunteer Romanian speakers (9 female, 8 male) reading isolated sentences from newspaper articles. 880 utterances were common to all speakers. This data was collected by the SWARA Project, funded by the Romanian Ministry of Education, for the purpose of building custom synthetic voices for individuals with speech impairments.

The IIIT-H Indic databases (Prahallad *et al.*, 2012) were collected for speech synthesis in a variety of Indic languages. One volunteer speaker per language read 1000 Wikipedia sentences selected for phonetic balance, resulting in about an hour and a half of speech per database. The languages and speaker genders are Bengali (male), Malayalam (male), Marathi (male), Tamil (male), Telugu (male), Hindi (female), and Kannada (female). Recordings are studio-quality. We examine the Telugu data in this work.

The MaryTTS project (Schröder and Trouvain, 2003) aims to make available TTS data and modular toolkits in many languages for teaching and research. They have made many of these tools and resources available online. In particular, we examine the MaryTTS Turkish corpus[1], which consists of about two hours of data from a male speaker.

3.3.1.2 Professional TTS Corpora

We were very fortunate to obtain production-quality, professional TTS data from three US English female speakers from Raul Fernandez and Andrew Rosenberg at IBM Research. This database consists of 1000 utterances comprising around an hour and a half of speech per speaker, and has formerly been used to create production-quality unit-selection text-to-speech systems at IBM.

[1] `https://github.com/marytts/dfki-ot-data`

The other TTS corpora we are looking at are freely-available TTS research corpora – thus, the speakers are typically not professional speakers. We were interested to compare production-level TTS data with research TTS corpora, and to see whether professional TTS data differs substantially, and to also compare with other professional speech (radio broadcast news). For these comparisons we focus on female speakers speaking US English.

3.3.2 Other Professional and High-Quality Speech

The Simple4All Tundra Corpus (Stan *et al.*, 2013) consists of approximately 60 hours of speech from 14 audiobooks, each in a different language, and each read by a single speaker (8 male, 6 female). It was collected for the purpose of providing found data in many languages for text-to-speech research. We examine the Romanian audiobook data, which is read by a female speaker, since we also have Romanian female TTS data for comparison.

Additionally, we obtained US English audiobooks, one male[2] and one female[3], from the LibriVox project website, as well as one audiobook in Telugu read by a male speaker[4].

The Boston University Radio News Corpus (BURNC) (Ostendorf *et al.*, 1995) is a corpus of professionally read radio broadcast news data and includes speech from seven (four male, three female) FM radio news announcers associated with the public radio station WBUR. The main corpus consists of over seven hours of news stories recorded in the station's studio during broadcasts over a two-year period. In addition, the same announcers were recorded in a laboratory setting where they read 24 stories from the radio news portion, first in a normal, non-radio style and then, 30 minutes later, in their radio style. We examined the broadcast radio news part of the corpus for our experiments here.

The Turkish Broadcast News Speech and Transcripts Corpus (Saraçlar, 2012) consists of about 107 hours of Voice of America radio broadcasts in Turkish, with corresponding transcriptions. There are four main male and three main female news anchors. We focus primarily on a clean subset of the data which has utterances containing background noise, music, or low-quality audio removed.

[2] `https://librivox.org/alices-adventures-in-wonderland-abridged-version-3-by-lewis-carroll/`

[3] `https://librivox.org/alices-adventures-in-wonderland-by-lewis-carroll-4/`

[4] `https://www.dasubhashitam.com/ab-title/ab-mangayya-adrushtam`

3.3.3 ASR Corpora

The CALLHOME corpus (Canavan *et al.*, 1997) consists of spontaneous, orthographically transcribed telephone conversations between native speakers of US English. The data includes 6 hours and 45 minutes of utterances from 86 different female speakers, 1 hour and 43 minutes from 32 male speakers, and 8 hours and 32 minutes from speakers whose gender was not annotated in the corpus.

The MACROPHONE corpus (Bernstein *et al.*, 1994) was designed for the development of telephone-based dialogue systems, such as travel booking and other database-related tasks. The utterances were read by 5,000 speakers over the phone. The data includes speech from male and female adults and children. We restricted our study to adult speakers of known gender.

The IARPA BABEL program (Harper, 2011) focused on the rapid creation of spoken keyword search systems for a diverse set of languages which have historically not received a great deal of attention from the speech research community (low-resource languages). While the goal of BABEL was primarily a speech recognition and spoken keyword search task, we are currently using some of this multi-speaker, conversational telephone data collected in 25 different languages for BABEL to build TTS voices for these languages. This data consists of both scripted and conversational telephone speech data from a variety of low-resource languages; in this work, we focus mainly on the languages for which we have corresponding data in other genres for comparison (namely, Telugu and Turkish). The unscripted speech was recorded from a variety of native speakers conversing over the telephone in a variety of noise conditions. The scripted speech was similarly collected from native speakers over the telephone in a variety of noise conditions, and consists of short read phrases and sentences typical of database-query tasks.

We also examined a Romanian read ASR corpus obtained online[5], which consists of speech from multiple male and female speakers.

For all of these ASR corpora, we examine features for 5 male and 5 female speakers, chosen randomly, for readability of the graphs, since there are hundreds of speakers in each of these corpora.

[5] `http://rasc.racai.ro/index.php?page=download`

3.4 Features, Tools, and Methods

We used Praat (Boersma, 2001) to extract maximum f0, commonly used as an estimate of pitch range (Hirschberg and Beckman, 1994), as well as mean energy, noise-to-harmonic ratio (NHR), jitter, and shimmer, for each utterance. We then aggregated these statistics for a given speaker or corpus by obtaining the mean and standard deviation for these values over all utterances for that speaker or corpus.

Also for each utterance we computed speaking rate in syllables per second using a Praat script for approximating syllable nuclei (de Jong and Wempe, 2009). While many TTS corpora contain syllable alignments that would enable more precise computation of speaking rate, and Festival TTS frontends are available as well for syllabifying text transcripts in many languages, much of our data does not contain this information (or a TTS frontend is not currently available in the language), so in the interest of consistency across corpora, we use the acoustic-only approach on all of our data.

Finally, we compute an articulation level for each utterance, which we define as (mean energy / speaking rate) * standard deviation of f0, such that a high articulation value corresponds to loud, slow speech with a large variation in pitch. High levels of articulation are commonly referred to as "hyper-articulation," and the opposite is known as "hypo-articulation." We likewise compute mean and standard deviation of articulation level for a speaker by computing mean and standard deviation of the individual utterance articulation values, for each of that speaker's utterances.

3.5 Results

3.5.1 Acoustic Features

According to (Matoušek *et al.*, 2008), TTS speakers are typically instructed to speak as consistently and with as little variation as possible. We do in fact observe that all research TTS corpora in each language show a lower maximum pitch, on average, by utterance, than other corpora in the same language and gender (Figure 3.1. For this study not all genres were available for all languages and genders, so only available ones are shown; error bars show +/- 1 SD). This is consistent with the expectation that speech collected for TTS will have less range and variation in pitch, as well as with our experimental findings (Cooper *et al.*, 2016b) that training a voice on a subset of the lowest mean f0 utterances produces a voice that is preferred by listeners over the baseline. We were nevertheless surprised that the professional TTS data (which we only have for US English) showed *higher* maximum pitch values, matching more with the audiobook style speech.

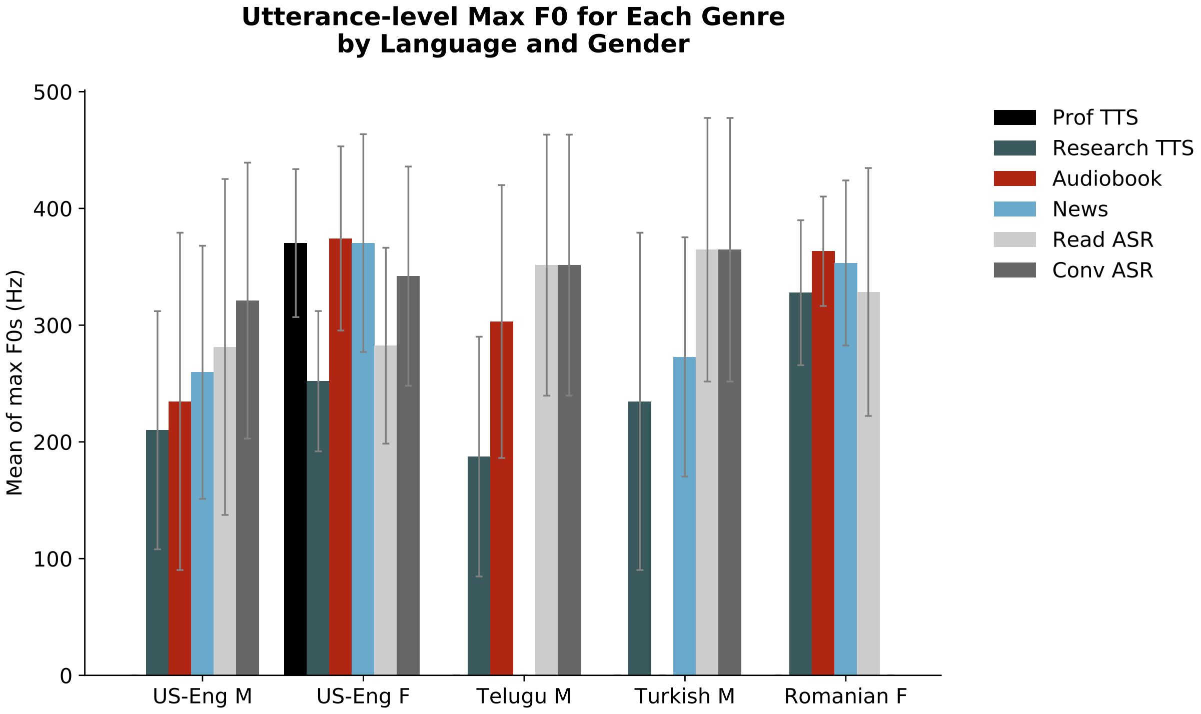


Figure 3.1: Research TTS corpora exhibit lower max pitch than other genres.

With respect to energy (Figure 3.2), as expected, the TTS corpora and broadcast news have lower standard deviations of energy, whereas among ASR corpora, there is a larger spread, and many speakers have a substantially higher standard deviation in energy. Audiobooks are somewhat close to broadcast news and TTS data in the graph, however the slightly higher standard deviation of energy is characteristic of the more expressive speaking style that we would expect in audiobooks. Again we can see that the three professional TTS speakers are somewhat closer in energy characteristics to the audiobook speakers. We would also note that energy is a characteristic both of the speaker and of the recording conditions; while it may be interesting to explore these separately in future work, we are interested in both, since they are both relevant forms of variation that we may encounter in different data sources.

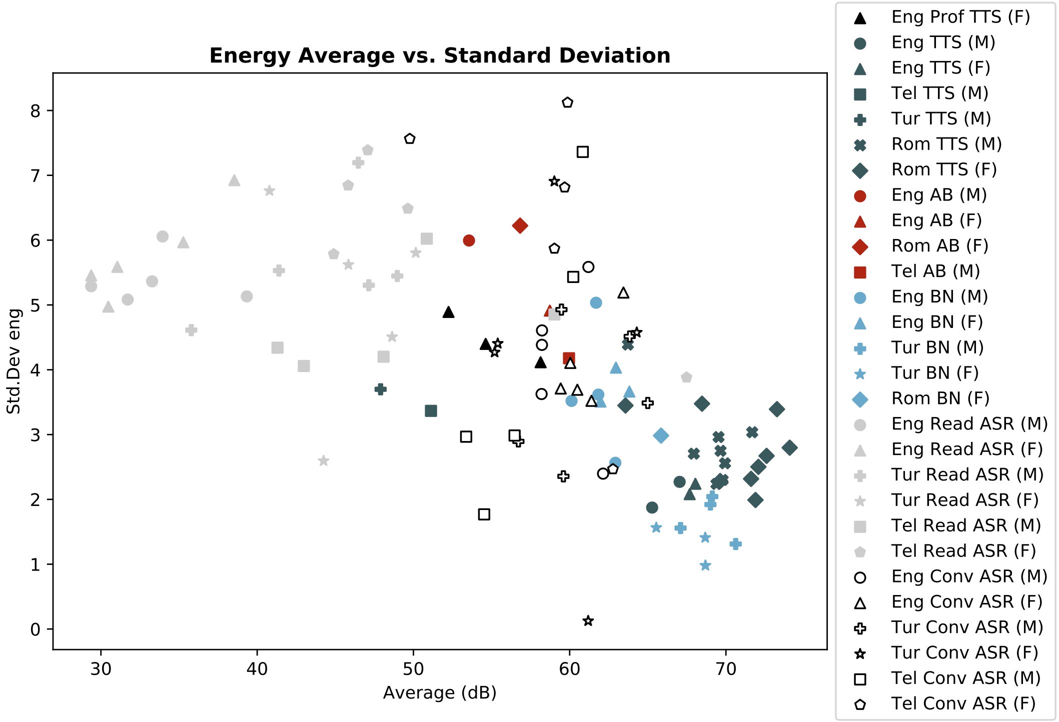


Figure 3.2: Broadcast news and TTS corpora have lower standard deviations for energy.

3.5.2 Speaking Rate and Articulation

For speaking rate (Figure 3.3), we see again that TTS and broadcast news speech are similar on average, but in terms of standard deviation, TTS speech has more variation. Audiobook speech is similar but slightly slower. It is apparent that some speakers in ASR corpora are similar in speaking rate to TTS speakers, whereas other speakers are very different.

We define a high articulation level as high mean energy, slow speaking rate, and high standard deviation of f0. We see once again (Figure 3.4; some outliers were removed for readability) that TTS data, broadcast news, and audiobooks generally cluster together at low values for average and standard deviation of articulation level. We see more spread among ASR speakers. Comparing TTS and audiobook data more directly by language and gender (Figure 3.5), it can be seen more clearly that audiobook speech has overall higher levels of articulation than the corresponding research TTS data. This matches our expectations of the more expressive style of audiobook speech, and indicates that if we wish to use audiobook speech for training TTS voices, it would be best to pick

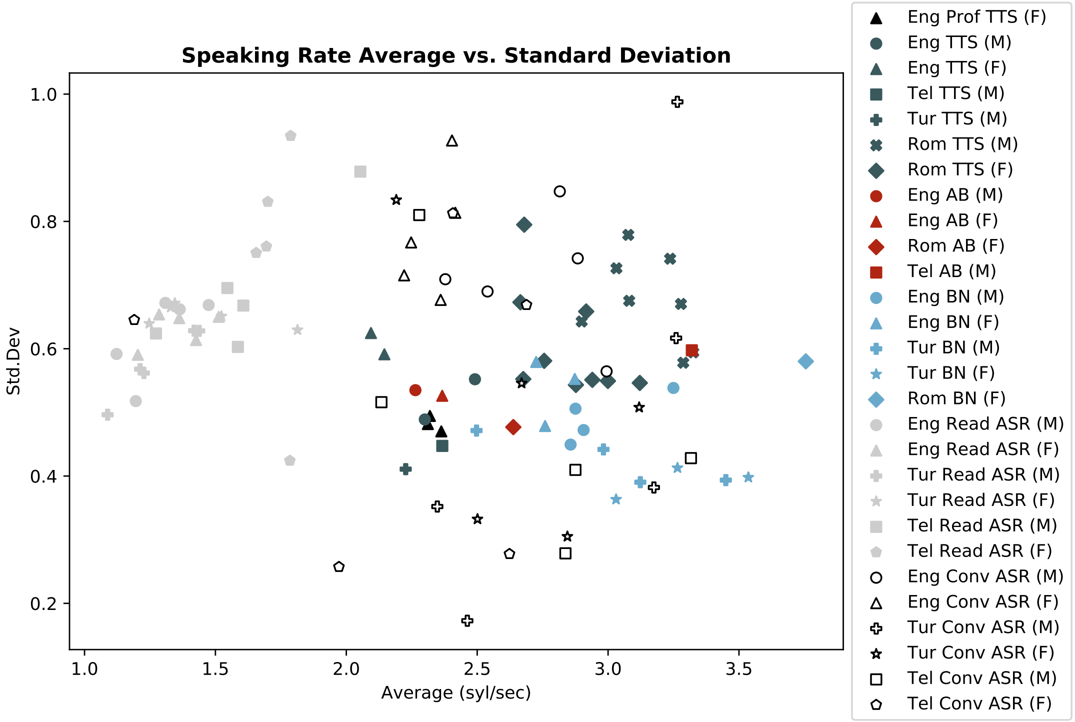


Figure 3.3: Audiobooks, broadcast news, and TTS corpora have similar speaking rates.

audiobook data that has lower levels of articulation, if we wish to match the "standard" TTS style. On the other hand, if we wish to intentionally create TTS with a more expressive style than the standard neutral TTS style, then audiobook data is an appropriate choice.

In previous work, we have consistently found that selecting less-articulated utterances for training TTS voices on found data produces better voices, both in terms of naturalness and intelligibility, likely because this data is more similar to TTS data (Cooper *et al.*, 2016b; Cooper *et al.*, 2016a; Cooper *et al.*, 2017). While audiobook data is promising for creating more expressive voices, this expressivity must be modeled appropriately if natural and intelligible expressive voices are to be created.

3.5.3 Voice Quality

Jitter, shimmer, and noise-to-harmonic ratio (NHR) are all features related to voice quality. Jitter is the cycle-to-cycle variation in f0, or in other words, frequency perturbation. Shimmer is cycle-to-cycle variation in amplitude, or, amplitude perturbation. NHR is the proportion of noise to

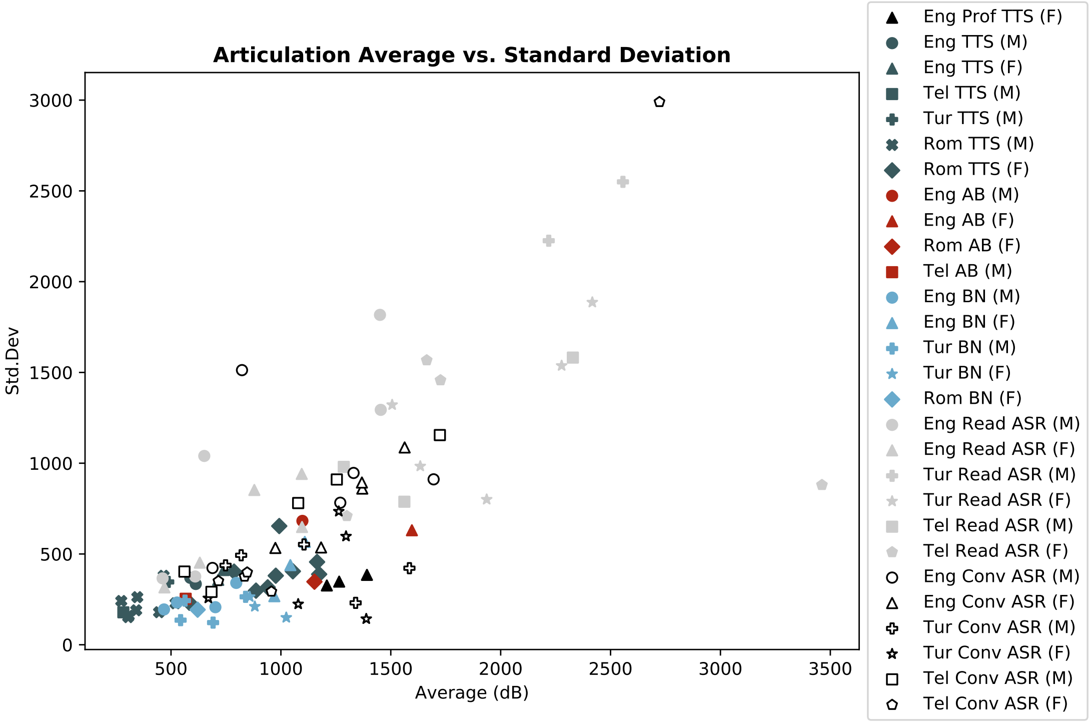


Figure 3.4: Level of articulation by genre.

the harmonic sounds in the voice. Jitter, shimmer, and a high NHR typically correspond to a hoarse, harsh, or rough-sounding voice. For these three features, we expected that TTS and other professional speech would tend to have low values for all of these, but in fact the averages for these genres were very spread out. The main difference that we did observe, however, was a gender difference – female speech tended to have lower means and standard deviations overall (Figures 3.6, 3.7, and 3.8; female data points are darker and male data points are lighter). Once we split the data by gender, we do see that TTS speech from female speakers does tend to have lower shimmer and NHR values than female speech from other genres, as expected.

3.6 Discussion

We have found that research TTS speakers do tend to follow the expected style for TTS speech. TTS speakers tend to have lower f0 range as measured by max f0, as well as lower standard deviation of energy, but not speaking rate. Furthermore, when we separate the data by gender, we

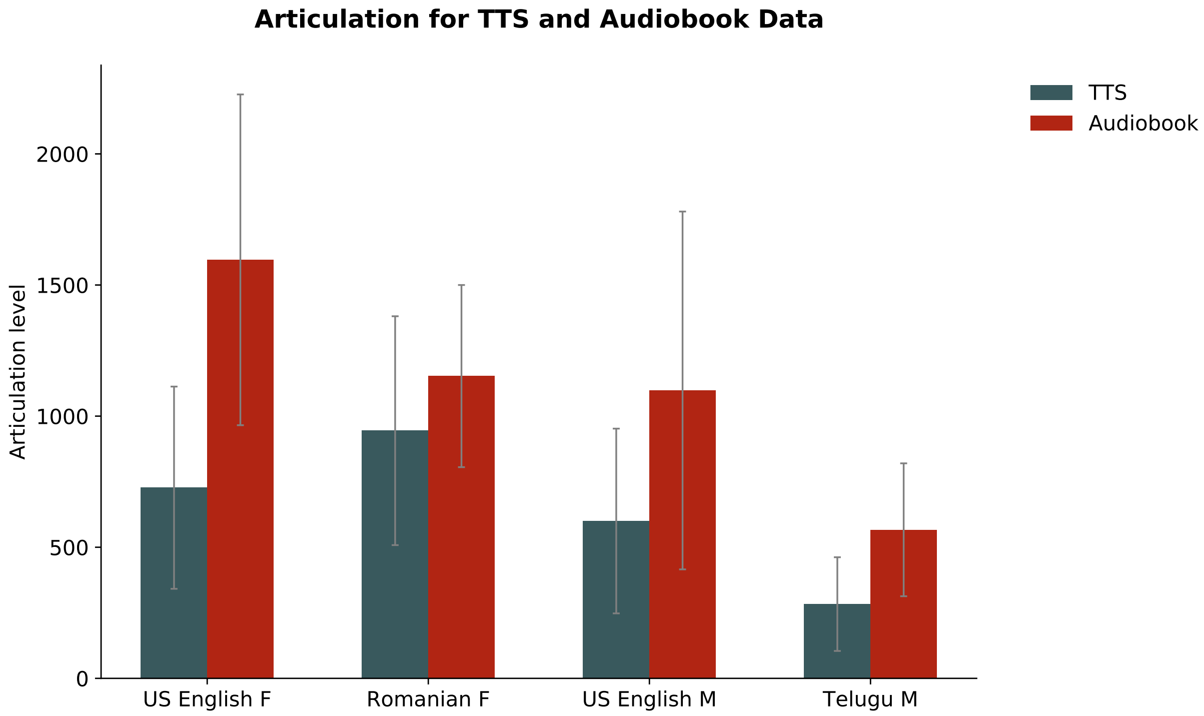


Figure 3.5: Audiobooks have higher levels of articulation than standard TTS data.

do observe that TTS speech has lower values on average for the voice quality features of shimmer and NHR. We observe that a major difference between TTS speech and audiobook speech is level of articulation, since the expressive style of reading that is characteristic of audiobooks tends to be louder, slower, and have more variation in f0. We also observe that a number of speakers in ASR corpora may be similar to TTS speakers in terms of these various acoustic and prosodic features, which may make these speakers more suitable for TTS than ASR speakers who are not similar in these features. We observe that professional TTS speakers also have a more expressive style, indicating that perhaps they are better able to produce expressive speech that still conforms to the requirements for TTS. These findings suggest objective justification for building TTS systems from particular "found" genres but also indicate the criteria that should be used to select data from other, less similar corpora, for building TTS systems, in the case where that is all that is available.

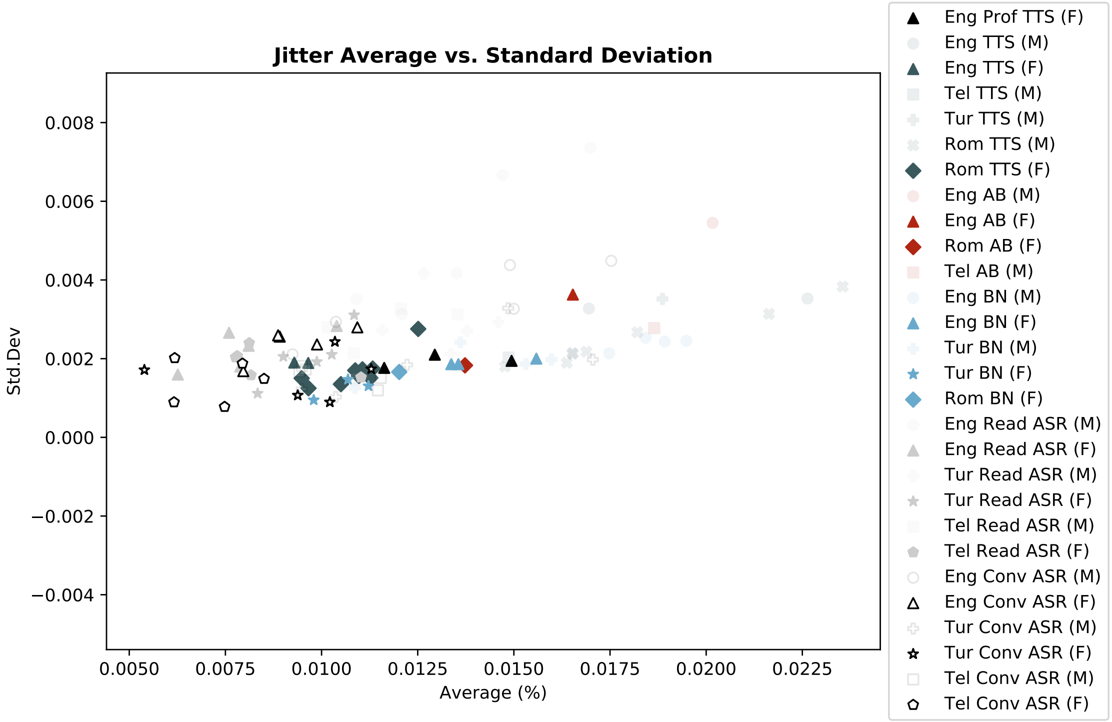


Figure 3.6: Female speakers tend to have lower average values for jitter.

3.7 Conclusions

In this work, we have identified features that characterize TTS corpora as well as which "found" corpora are most similar to TTS data – radio news and audiobooks. In so doing, we have also identified which features are important to use when choosing subsets of utterances from the less similar genres we examined, such as conversational corpora. Thus, not only can we demonstrate *why* certain "found" corpora are particularly well adapted for TTS voice construction but we can predict, from empirical findings, what subsets of other corpora should be either included or excluded from TTS voice construction.

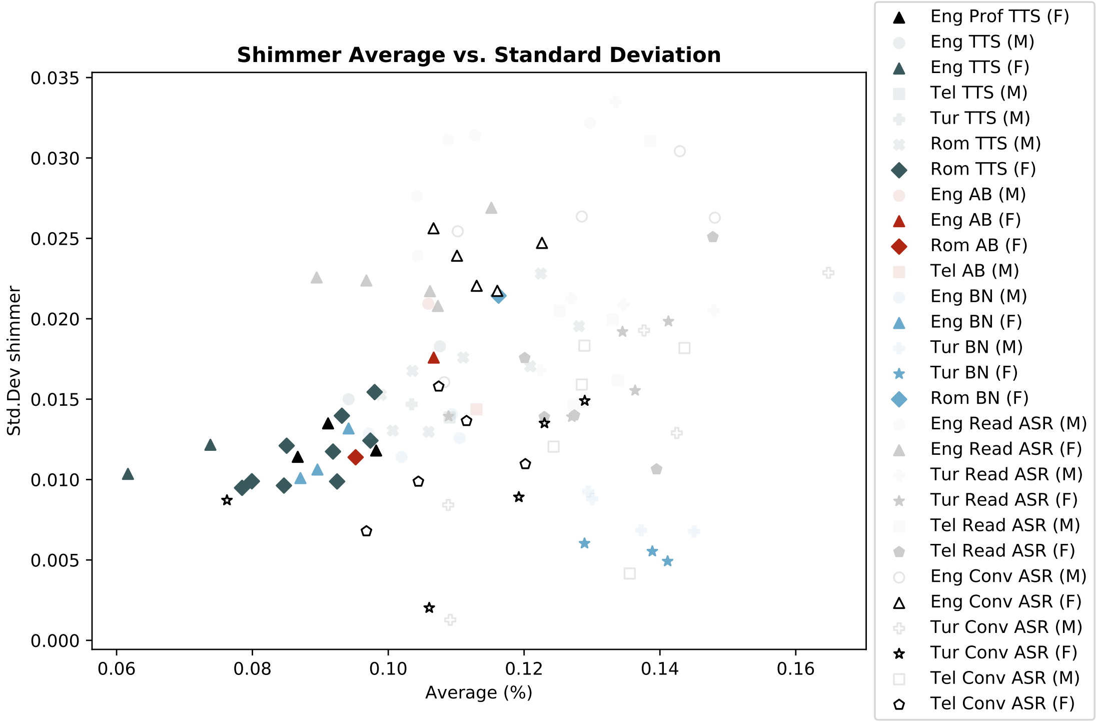


Figure 3.7: Female TTS speakers tend to have low values for jitter on average.

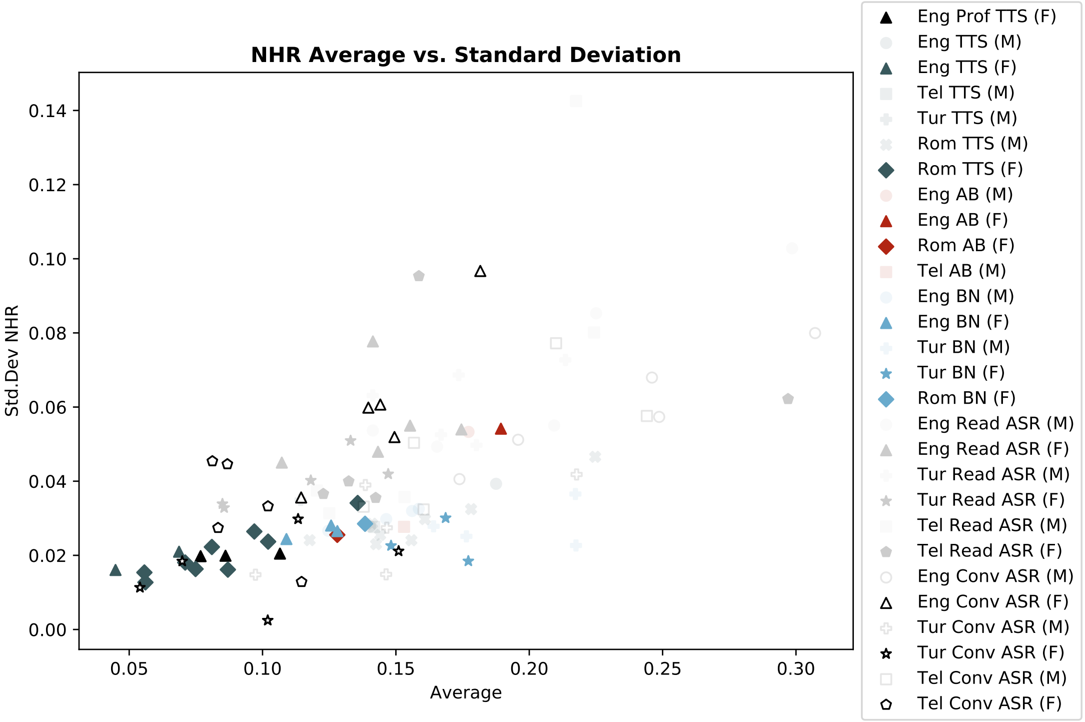


Figure 3.8: Female speakers tend to have lower means and standard deviations for NHR.

Chapter 4

Subset Selection for Intelligibility

4.1 Introduction

In order to train intelligible voices on found data collected under various conditions, we must identify ways of selecting the most appropriate portions of the corpus for voice training, while excluding data that introduces excessive noise and other artifacts. Found data, and in particular data collected for training automatic speech recognition (ASR) systems, presents the challenge of containing informal conversational speech, many types of background noise, and large numbers of speakers. Indeed, the goals when training a speech recognizer are often orthogonal to the requirements for training a TTS system – a recognizer needs to be able to function regardless of noise, recording conditions, or speaker style and characteristics. We conducted a number of experiments training voices on data collected for ASR, and informal listening tests on these voices revealed that intelligibility is the main challenge when training voices on this data. So, in these experiments we evaluate and optimize for intelligibility. We select training data at the utterance and speaker level based on a number of different criteria which we hypothesize will produce more intelligible and natural-sounding voices, such as speaking rate, f0, energy, and absence of noise.

The basic training data subset selection approach is as follows:

- Label each utterance (or, all the data from one speaker) with its value for some given feature (e.g., speaking rate).
- Sort all utterances (or speakers) on that feature, from low to high.

- Select low, middle, or high-valued subsets for that feature by starting at one end of the list (or, the middle of the list) and accumulating utterances one by one until you have a set containing the desired amount of audio (e.g., 2 hours).

4.2 Related Work

Previous work on selecting the best data from a noisy or non-homogeneous corpus has typically involved removing the noisiest utterances and choosing the most neutrally-spoken portions of the data. Audiobooks and radio broadcast news have been popular sources of found data (Chalamandaris *et al.*, 2014; Watts *et al.*, 2013; Braunschweiler and Buchholz, 2011; Gallardo-Antolín *et al.*, 2014) due to their relatively clean recording conditions and the fact that they typically contain large amounts of speech from a single speaker. Corpora designed for automatic speech recognition have also been explored for building HMM-based TTS voices; in particular, (Yamagishi *et al.*, 2010) built TTS voices on various ASR corpora containing cleanly-recorded read speech, as well as some corpora containing speech in a noisy environment, with the goal of being able to create "thousands of voices" from the many speakers in each corpus. (Baljekar and Black, 2016) identified noisy and misaligned utterances in a corpus of conversational telephone speech by measuring mel cepstral distance (MCD) between original utterances and utterances synthesized by a model trained on all of the data, in order to find utterances that are outliers with respect to the overall data.

Most work on building TTS from found data focuses on utterance-level selection. There has however been some work in selecting the best *speakers* for building a voice. (Gutkin *et al.*, 2016) identified speakers that were acoustically similar to each other for building a Bangla voice. They auditioned 15 speakers, did a crowdsourced evaluation to identify the speaker most preferred by listeners, and then picked five additional speakers from the original 15 who had the most similar vocal characteristics.

While prior work has begun to investigate the use of found data for TTS, many questions remain including the best ways to filter data from sources such as ASR corpora. In our work, we aim to identify utterances and speakers in ASR data that are both similar to each other and suitable for TTS, by selecting speakers based on a number of novel acoustic and prosodic features, chosen because they are language-independent and simple to compute. We also aim to compare utterance-

and speaker-level data selection to determine which produces better voices.

4.3 Voices Trained on Conversational Telephone Speech: CALLHOME

CALLHOME (Canavan *et al.*, 1997) is representative of data typically collected for ASR — noisy, informal, conversational telephone speech from a multitude of speakers. Although it is US English, it is similar to the conversational portion of the BABEL data (Harper, 2011) in low-resource languages. While this type of data is likely to be among the first collected for a new language, its heterogeneity presents a challenge for training TTS voices. We explore approaches for selecting subsets of the data based on acoustic and prosodic features, as well as creating subsets by removing the types of utterances that we hypothesize may hurt the intelligibility of the voice. We trained voices using only the 6 hours and 45 minutes of data labeled as being spoken by female speakers in order to produce more consistent models.

The CALLHOME data was transcribed as two-sided conversations; for the purpose of these experiments, an utterance was defined as a conversational turn. Our baseline was a voice trained on *all* 6 hours and 45 minutes of the female utterances (no subset selection). This produced a voice that had a word error rate of **32.9%** when transcribed by Mechanical Turk workers. Training a voice on all of the data, including male, female, and unknown-gender speakers, produced a voice with a WER of 49.4%, justifying our choice to use only female-labeled data.

We trained our TTS voices using the Hidden Markov Model Based Speech Synthesis System (HTS) (Zen *et al.*, 2007). We based our training recipe for the baselines and for the data selection subsets described above on the speaker-independent training recipe, described in detail in Chapter 2, Section 2.2. We treat all of the data in each subset as if it were from one speaker, primarily for computational efficiency, as we wanted to be able to create many voices as rapidly as possible. We obtained the standard set of full-context phonetic labels using the Festival Speech Synthesis System (Black *et al.*, 2014). We measured intelligibility by obtaining transcriptions on Amazon Mechanical Turk for semantically-unpredictable sentences spoken by each voice.

4.3.1 2-hour Subsets Chosen from Acoustic and Prosodic Features

As a starting point, we produced 2-hour subsets of the female CALLHOME data based on a number of acoustic and prosodic features: mean and standard deviation of energy and fundamental frequency (f0) as computed using the Praat software (Boersma, 2001), and speaking rate defined as syllables per second. We sorted all of the utterances based on each feature, and then selected the top, middle, and bottom 2 hours of data to produce training data subsets. We trained a voice on each subset, and then sent samples of each voice to be evaluated on Amazon Mechanical Turk. Intelligibility results can be found in Table 4.1.

Feature	WER %
Baseline	*32.9*
Low mean f0	65.4
Middle mean f0	72.6
High mean f0	74.9
Low stdv f0	69.4
Middle stdv f0	73.1
High stdv f0	75.4
Low mean energy	66.6
Middle mean energy	74.3
High mean energy	52.6
Low stdv energy	72.9
Middle stdv energy	71.4
High stdv energy	64.0
Slow speaking rate	96.6
Middle speaking rate	46.9
Fast speaking rate	66.3

Table 4.1: Word error rates for voices trained on 2-hour subsets, selected using acoustic and prosodic features.

Although none of these test voices improve over the baseline, the "slow speaking rate" voice stands out as the *least* intelligible, and “middle speaking rate” is the most intelligible, indicating that this speaking rate of the training data plays an important role in intelligibility of the output voice. One might expect that slower speech should be *more* intelligible; however, the conversational nature of the CALLHOME data means that the utterances with the slowest average speaking rate often contain backchannels, hesitations, or disfluencies, all of which are not ideal for building voices.

4.3.2 1-hour and 4-hour Variants of Best Voices

Looking at the subsets that did relatively better, we next varied subset sizes to 1 hour and 4 hours. For our five best-performing subsets (middle speaking rate, high mean energy, high standard deviation of energy, low mean f0, and fast speaking rate), we created these additional subsets; results are shown in Table 4.2. 2-hour subset voice results are repeated for comparison. The three best voice results appear in bold.

Feature	1hr	2hrs	4hrs
Middle speaking rate	79.7	**46.9**	51.4
High mean energy	77.4	52.6	55.4
High stdv energy	64.9	64.0	**50.6**
Low mean f0	75.4	65.4	52.3
Fast speaking rate	91.1	66.3	**38.3**

Table 4.2: Word error rates for voices trained on subsets of different sizes of our five best acoustic and prosodic features.

Here we find that the one-hour subset size does consistently worse that the others, indicating that we should generally be using larger subsets with this type of data.

4.3.3 Removal of Noisy Utterances

Conversational telephone speech data contains noise in many forms, which would not be present in a corpus of professional speech, and which is likely to hurt the quality of voices trained on that data. We thus considered various methods of identifying and removing noisy utterances.

First, we created one subset that excluded any utterances containing disfluencies that we could identify in the transcripts – filled pauses such as “um” and “uh” — as well as word fragments, indicated by hyphens. Then, we created a second 2-hour subset of the utterances with the lowest signal-to-noise ratio (SNR), computed using the SNR tool from the NIST Speech Quality Assurance (SPQA) package (NIST, 2009). Finally, we observed that many of the very shortest utterances were backchannels, often spoken slowly (such as "yeah" and "uhhuh"), which were possibly degrading the overall quality of the synthesized voice. We thus created a third training set that excluded single-word utterances, and a fourth set that excluded all utterances of one or two words. Finally, since clipping distorts the audio, we created a subset of utterances with no clipping, measured using the `afclip` command line utility provided with Mac OS X. Results for these voices can be found in Table 4.3, which also shows the size of each subset in hours.

Filter	Hours	WER
Baseline	*6:45*	*32.9*
No disfluencies	5:03	49.4
SNR	2:00	80.3
2 or more words	6:25	43.1
3 or more words	6:11	37.4
No clipping	6:21	52.0

Table 4.3: Word Error Rates for different methods of removing noisy utterances.

While, rather surprisingly, none of these methods of noise removal resulted in a more intelligible voice than the baseline, removing short utterances appears to be one of the more promising approaches.

4.3.4 Utterance Clustering

Based on the hypothesis that utterances from similar speakers will create a better voice due to more consistent models, we used the LIUM speaker diarization tool (Rouvier *et al.*, 2013) to cluster our dataset (11094 utterances concatenated into a single audio file of 6 hours and 45 minutes) into groups that seemed to belong to a single hypothesized "speaker"; in actuality, the utterances in each cluster are from multiple speakers. LIUM grouped our utterances into 75 clusters based on speaker similarity, and we chose the utterances corresponding to the largest cluster (5 hours and 13 minutes) and trained a voice using the utterances in that cluster. This voice gave a WER of 58.9%. Due to our observation that shorter utterances were mainly backchannels which may be inappropriate to include as training data, we next tried clustering only the longer utterances: we used the 5 hours and 34 minutes of longer utterances (excluding the ones that are only a few milliseconds) and clustered using LIUM again. This time, it clustered our data into 51 clusters, and we trained a voice on utterances from the largest cluster again (4 hours and 24 minutes). This voice gave a WER of 38.9%. We see a huge improvement in WER by cutting out the shortest segments, indicating that these utterances may indeed be hurting the quality of the voice.

4.4 Voices Trained on Read Telephone Speech: MACROPHONE

The MACROPHONE corpus (Bernstein *et al.*, 1994) consists of short read utterances over the telephone in US English, such as names, dates, and places. This data is similar to the read portion of the BABEL data, and we hypothesize that the read style will be more clear and consistent than informal, conversational speech and thus better-suited to training TTS voices. We continued to train our voices using the Hidden Markov Model Based Speech Synthesis System (HTS) (Zen *et al.*, 2007), using the speaker-independent recipe, and we evaluated the voices for intelligibility on Amazon Mechanical Turk via the transcription task using semantically-unpredictable sentences. We also explored automatic evaluation of voices using automatic speech recognizers. This work has been published in (Cooper *et al.*, 2017).

4.4.1 Utterance Selection Experiments

We started with just the first 10 hours out of the 83 hours of adult female-labeled utterances in this corpus, to enable a comparison with the 6 hours and 45 minutes of CALLHOME, using the same order of magnitude of data of a different type.

We selected our training subsets based on criteria such as mean and standard deviation of f0 and energy, as well as speaking rate (computed as syllables per second), level of articulation (computed as mean energy divided by speaking rate), and utterance length. For each feature, we computed its value for each utterance, and then sorted the list of utterances from low to high. Then, we obtained subsets by selecting e.g. the first two hours' worth of utterances from that list. We also experimented with removing different types of utterances that we hypothesized might hurt the quality of the voice, such as very short utterances of only one or two words; utterances containing clipped audio; utterances containing transcribed noise (such as "[unintelligible]," "[bg_speech]," and "[line_noise]"); and utterances consisting of a word spelled out letter-by-letter, which are indicated in the corpus by "spword" in the file name.

We trained our baseline voice on all of the first 10 hours of female utterances – training on the full set of over 80 hours would be prohibitively computationally expensive. This produced a voice with a word error rate of **67.7%** when transcribed by Mechanical Turk workers. We compare all of our MACROPHONE subset voices to this baseline. Results can be found in Tables 4.4 and 4.5. Voices that did better than the 10-hour baseline appear in bold.

Feature	WER %
Baseline	*67.7*
Low mean f0	98.6
Middle mean f0	85.7
High mean f0	100.3
Low stdv f0	83.1
Middle stdv f0	80.0
High stdv f0	87.1
Low mean energy	98.6
Middle mean energy	95.7
High mean energy	70.6
Low stdv energy	100.9
Middle stdv energy	85.4
High stdv energy	79.7
Slow speaking rate	96.6
Middle speaking rate	99.1
Fast speaking rate	**54.3**

Table 4.4: Word error rates for voices trained on 2-hour subsets of the first 10 hours of female MACROPHONE data, selected based on acoustic and prosodic features.

While we expected the MACROPHONE read telephone speech to produce more intelligible

Subset	Hours	WER
Baseline	*10:00*	*67.7*
3 or more words	7:34	79.7
No clipping	9:57	77.7
No transcribed noise	5:53	**58.9**
No spelled words	9:24	94.3

Table 4.5: Word error rates for voices trained on subsets of the 10 hours of data minus utterances removed based on different noise criteria.

voices than CALLHOME, this was surprisingly not the case: the CALLHOME baseline was rated as more intelligible than our MACROPHONE baseline. However, we are able to see approaches that give an improvement — namely, the two-hour subset of utterances with the fastest speaking rate and the subset that excludes utterances with transcribed noise that do beat the MACROPHONE baseline.

While in our initial MACROPHONE experiments we only used the first 10 hours of female data to enable comparison with CALLHOME, there is actually over 83 hours of female data available in the corpus. We hypothesized that being able to select even a small subset from a larger amount of data would give us access to potentially a larger pool of suitable utterances. We therefore repeated our data selection experiments by choosing 2- and 4-hour subsets of the entire 83 hours of female speech rather than just selecting from the first 10 hours. We excluded utterances containing transcribed noise, since we have established in our previous experiments that excluding these gives an improvement. We only repeated selection based on the 5 features that performed the best in our preliminary experiments: fast speaking rate, high mean energy, low level of articulation, high standard deviation of energy, and middle standard deviation of f0. Intelligibility results are in Table 4.6. Results better than our baseline (67.7%) appear in bold.

When looking at results for selecting 2-hour subsets from just the first 10 hours of data (Table 4.4), versus selecting 2-hour subsets from the full 83-hour data set (Table 4.6, first column), we noticed that selecting subsets from the full dataset does usually produce more intelligible voices than selecting just from the first 10 hours, with most of these voices being rated as more intelligible than the 10-hour baseline, despite being trained on only 1/5 of the amount of data. Extending to

Subset	2hr WER	4hr WER
Fast speaking rate	**66.6**	**48.3**
High mean energy	**60.0**	**48.3**
Low articulation	**64.6**	**49.1**
High stdv energy	83.1	**64.6**
Middle stdv f0	**48.0**	**45.1**

Table 4.6: Word error rates for voices trained on subsets of the full 83 hours of data.

4-hour subsets consistently produces better voices than the baseline as well. This indicates that more data is more likely to give an improvement if it is chosen in a principled way, and validates our hypothesis that better voices can be trained by identifying the best training utterances in a noisy corpus, even if this results in less training data.

Similar to CALLHOME, three of our best voices were created from fast speaking rate utterances – both the 4-hour and 2-hour sets selected from the full data set, and the 2-hour set selected from just the first 10 hours. A low level of articulation, which encodes fast speaking rate, also proved to be a preferable feature – voices trained on the 4 hours and 2 hours of most hypo-articulated utterances selected from the full data set scored better than the baseline. Our top two voices were based on selecting utterances with middle values for standard deviation of f0, with the 2-hour subset of the full data at 48% and the 4-hour subset at 45.1%. This was surprising as we would expect *low* values of f0 standard deviation to be more consistent with the speaking style in a standard text-to-speech corpus, according to (Matoušek *et al.*, 2008); however, corpora collected for TTS are typically optimizing for naturalness, generally in a unit selection setting, whereas here we are optimizing for intelligibility.

4.4.2 Automatic Evaluation Using Speech Recognizers

A limitation of our approach is the long turnaround time for crowdsourcing voice transcriptions. Since each worker may only transcribe one voice, evaluation proceeds very slowly. This led us to investigate the use of automatic speech recognition (ASR) to evaluate intelligibility. Although an ASR system will not interpret a voice exactly as a human would, depending heavily upon the type of data on which it was trained, it would nevertheless return results very quickly and not have the limitation of remembering and being influenced by repeat sentences. We therefore thought it would be worthwhile to see how this type of evaluation compares to that done by humans and whether in fact there are some reliable correlations.

We tested three different general-purpose, state-of-the-art, industry-level ASR APIs (Application Programming Interfaces) to determine the viability of their use for evaluating voices: wit.ai[1], a natural language API toolkit owned by Facebook; Watson[2], IBM's API for cognitive applications; and the Google Cloud Speech API[3]. We decided to try APIs rather than building our own ASR because using state-of-the-art recognizers should presumably provide the best possible proxy for a human listener. Our hypothesis was that some of these recognizers might correlate well with human transcription performance, so we could use these as a first step to choosing our best candidate voices to send to MTurk.

For each voice, we ran the same set of synthesized SUS that we gave to MTurk workers through each ASR API. We then computed WER from the returned transcripts. We allowed the same singular/plural and compound word confusions that we allowed in the transcriptions from Mechanical Turk, but we did not need to correct for spelling or typographical errors.

We found strong correlations between our three different ASR APIs' WERs and those from MTurk. We report correlations across our 34 different voices in Table 4.7. Furthermore, for all voices that humans rated as better than baseline, all three ASRs agreed that these voices were better than the baseline. This indicates that using ASR APIs is a promising pre-selection approach to decide which voices should get evaluated on MTurk.

While using ASR is much faster than crowdsourcing, it comes with its own challenges. For example, we noticed that sending the same audio clip multiple times to the same ASR did not necessarily always return the same transcription. A major downside of using an ASR API is that we have no information on the internal system — not just the type of models they are using, but how often they are updated, or whether some machines in their cloud are running different versions of the recognizer. So we can only speculate as to why repeated recognition of the same audio file would result in multiple different 1-best transcripts each time. We originally thought that using an

[1] `https://wit.ai/`; accessed March 1, 2017.

[2] `https://www.ibm.com/watson/developercloud/speech-to-text.html`; accessed March 1, 2017.

[3] `https://cloud.google.com/speech/`, accessed March 1, 2017.

ASR API would serve as a very consistent way of evaluating the voices, but this may not be the case.

Nevertheless, the human evaluations are also somewhat inconsistent and, in fact, they tend to be more inconsistent than the ASR evaluations. We have measured standard deviations in word error rate for the baseline MACROPHONE voice across the 5 workers who transcribed it, as well as standard deviations for those same utterances sent 5 times each to our three ASR APIs, as a way to measure variability of the different systems; these are also reported in Table 4.7. Furthermore, Figure 4.1 shows WER ratings for each of the four evaluation methods across our 60 voices.

Evaluation	Correlation (r)	Std.Dev (%)
MTurk	—	4.52
wit.ai	0.728	1.20
Watson	0.797	0.00
Google	0.876	0.00

Table 4.7: Correlation of ASR APIs with MTurk on 34 voices, and standard deviation in WER when evaluating the baseline MACROPHONE voice 5 times.

Both Watson and Google returned the same transcripts all five times, indicating that they have the least variability.

We have also found challenges related to task specification. While we were able to tell MTurk workers in the instructions that the sentences they were transcribing would not necessarily make sense, we could give no such instructions to an ASR system. We noticed that, for example, wit.ai appeared to be attempting to recognize sentences or parts of sentences that "made sense" – most likely because its language model was trained on semantically-predictable data. While wit.ai had the most obvious language model effects, this also applies to *any* ASR API over which we do not have control. Ideally we would be able to "tell" an ASR that the sentences will not necessarily make sense, e.g. by specifying a very simple language model such as a unigram or bigram, but unfortunately we have no such control over a cloud-based ASR API. Thus, for work with low-resource languages, we may wish to use our own ASR systems trained on the same data as the TTS voices, since in the case of actual low-resource languages, this may be all we have available. This will allow us complete control over the language model and over the system in general so that we

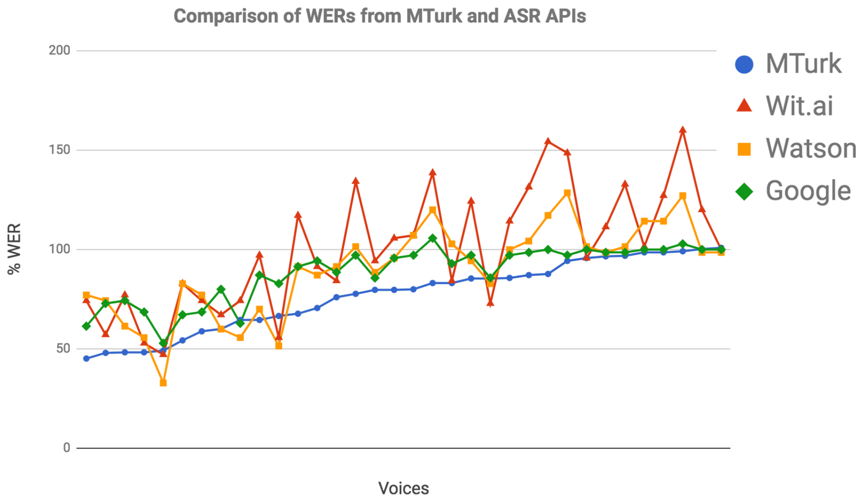


Figure 4.1: Comparison of WERs from Mechanical Turk and ASR APIs across 60 voices

can ensure consistent evaluation.

4.5 Utterance vs. Speaker Selection: MACROPHONE

We previously experimented with a data selection approach in which we attempted to identify a subset of *utterances* out of the entire corpus which would be the most suitable for building TTS voice models. Next, we extended those experiments by investigating data selection at the *speaker* level in addition to the utterance level; we aim to determine whether certain *speakers* in the corpus speak in a manner that is better suited to building voice models. We aim to identify speakers in ASR data that are both similar to each other and suitable for TTS, by selecting *speakers* based on various acoustic and prosodic features. We also aim to compare speaker-level training data selection to utterance-level training data selection in order to determine which method produces better voices. Our hypothesis is speaker-level selection, by using more data from fewer speakers, will produce more consistent models and thus more intelligible-sounding synthesized speech. This work has been published in (Lee *et al.*, 2018).

4.5.1 Tools and Methods

Between this set of experiments and the previous set, neural network based synthesis became the predominant method of acoustic modeling for TTS research, and the Merlin toolkit (Wu *et al.*, 2016) became available for public use. Preliminary informal experimentation revealed that Merlin produced more intelligible voices from ASR data than HTS did, so at this point, we switched to mainly using Merlin. Thus, to be able to directly compare speaker- vs. utterance-selected voices, we re-trained utterance-selected voices using Merlin.

Furthermore, because of the promising initial results of using ASR APIs for objective evaluation in the previous set of experiments, we initially evaluate all of our voices in this way, using the IBM Watson Speech to Text Service. Then, we only send our best five voices from each selection method to Mechanical Turk for human evaluation for intelligibility using our transcription task.

Since we can initially evaluate and eliminate candidate voices more quickly and easily using a preliminary objective evaluation, we decided to expand the set of features we are exploring for training data subset selection. Praat (Boersma, 2001) enables extraction of many different acoustic and prosodic features, including the following standard ones:

- F0 (min, max, mean, median, and standard deviation)
- Mean absolute f0 slope (MAS)
- Energy (min, max, mean, and standard deviation)
- Ratio of voiced to total frames (vcd2tot)

We also calculated speaking rate in syllables per second based on the syllable information in the label files obtained with Festival. We measured level of articulation, which we defined as mean energy divided by speaking rate, to encode the loud and slow speech that characterizes hyper-articulation. Finally, we obtained an approximate measure of speaker intelligibility, by running their speech through the Watson ASR API and calculating a WER for each speaker. We computed the same features at both the utterance level as well as the speaker level, in order to compare speaker- vs. utterance-level data selection.

4.5.2 Experiments and Results

Our baseline voice was the first 10 hours of female Macrophone data, with no feature-based data selection. Our test voices were trained on 2-, 4-, and 10-hour subsets chosen by these features and selected out of the entire 83 hours of female data. For example, for the 4-hour high-clustered mean f0 speaker-selected voice, we sorted all female speakers by their mean f0 and accumulated data from the speakers by their distance from the highest mean f0 speaker one by one until we had a training set containing 4 hours of data. We trained voices for low, median, mean, and high-clustered values of each of our 14 features for both speaker-selected and utterance-selected subsets. After observing that our best speaker-selected voices were all trained on 10 hours of data as opposed to smaller amounts, we then only trained 10-hour voices for utterance selection.

As previously mentioned, we used the "build your own voice" recipe from the Merlin toolkit to train all of our voices. We performed a standard 10:1:1 train/development/test set split for all voices trained.

Due to the slow turnaround time for evaluating synthetic voices for intelligibility by human transcription, we did an initial automatic evaluation of all of our voices using the Watson ASR API, which we have found in our prior work to correlate well with human judgments for intelligibility of synthetic voices (Section 4.4.2). Synthesized test utterances consisted of eleven 7-word semantically-unpredictable sentences (SUS) of the standard form `det adj noun verb det adj noun`, in order to prevent contextual recognition of words. After obtaining a set of hypothesized transcriptions, we computed WER for each voice. Our baseline voice had an automatically-obtained WER of **82.9%**. 74 out of the 168 total speaker-selected voices, and 28 out of the 56 total utterance-selected voices, had smaller WERs than the baseline voice. For brevity, the top 5 voices for each selection method, with their WERs from Watson, are in Tables 4.8 and 4.9. Full results can be found in Appendix B.

Next, it was necessary to corroborate our automatic evaluation with human judgment. This is especially important for the voices trained on subsets selected based on low ASR WER; we would expect those voices to do well when evaluated by the same ASR, as they did, but we need to verify whether humans perceived the best automatically-rated voices as being high quality as well. We created an intelligibility task on Mechanical Turk using our baseline voice, our 5 best speaker-selected voices, and our 5 best utterance-selected voices, as well as one semantically-predictable sentence spoken clearly by one of the authors, intended as an attention check for MTurk workers.

Speaker feature	Cluster	WER
WER	Low	58.4%
Voiced vs. total	High	59.7%
Mean energy	Low	61.0%
F0 MAS	Avg	61.0%
Min energy	Low	62.3%

Table 4.8: Watson WERs for 5 Best Speaker-Selected Voices

Utterance feature	Cluster	WER
Mean energy	High	67.5%
Mean F0	Median	68.8%
Mean energy	Median	71.4%
Max F0	High	74.0%
Speaking rate	Low	74.0%

Table 4.9: Watson WERs for 5 Best Utterance-Selected Voices

We synthesized the same set of 11 SUS with each of these 11 voices, and we presented them to workers in a Latin-square configuration. WER results as well as p-values from a two-tailed t-test in comparison with the baseline are reported in Table 4.10.

We observed that our 5 best speaker-selected voices were all rated as more intelligible than all 5 of our best utterance-selected voices. Furthermore, all speaker-selected voices were rated as more intelligible than the baseline, with the top four obtaining significance at $p < 0.05$. The best performing voice was trained on the 10 hours of data from the speakers with lowest WER as determined automatically by Watson ASR, indicating that this ASR matches well with human perception of speech for intelligibility, and that selecting more intelligible speakers for training data does produce more intelligible voices. Training on speech with a greater proportion of voiced data also produced a more intelligible voice, and lower energy levels appeared to be a useful selector at the speaker level as well. It would be interesting to know whether the training data subsets that produced better voices consisted of fewer speakers with more data; further analysis of the characteristics of the subsets that produced the best voices is a possible direction for future work.

Unit	Selection Feature	Cluster	WER	*p*-value
Spkr	WER	Low	**41.8%**	**0.0145**
Spkr	Voiced vs. total	High	43.4%	0.0175
Spkr	Mean energy	Low	43.4%	0.0216
Spkr	Minimum energy	Low	43.6%	0.0275
Spkr	F0 MAS	Average	46.2%	0.1037
Utt	Mean F0	Median	54.3%	0.9618
	[Baseline]		54.5%	
Utt	Mean energy	Median	55.3%	0.8793
Utt	Mean energy	High	57.7%	0.5354
Utt	Speaking rate	Low	62.9%	0.0882
Utt	Max F0	High	63.1%	0.0971

Table 4.10: MTurk Results for Single-Feature Utterance- and Speaker-Selected Voices

Comparing the features that did well for utterance selection to features that did well for speaker selection, it is interesting to note that they are different. For instance, Watson WER was *not* in the top 5 features for utterance selection. This indicates that selecting training data based on which *speakers* are most intelligible is a good approach, but training on only the most intelligible *utterances* regardless of speaker perhaps does not result in as cohesive a training set. Looking at which best features are common across utterance and speaker selection, mean energy stands out as appearing in both sets, with low mean energy being a good selector for speakers, and both middle and high mean energy appearing to be good selectors for utterances. This indicates that perhaps the actual energy level is not so important, but that having a similar energy level across your training data will produce a better voice.

4.5.3 MCD for Evaluation

Because low-resource languages do not necessarily have high-quality or reliable ASR, we experimented with the mel-cepstral distortion (MCD) objective function (Kubichek, 1993). This function measures the difference between two time-aligned mel-cepstral sequences and is commonly used as an objective evaluation metric for TTS. For unaligned sequences, dynamic time warping may be performed to align the sequences before comparing them.

We computed average MCD on the test set of each voice and found an 0.701 Pearson correlation between MCD and IBM Watson WER among all voices; correlation increased to 0.756 when considering only the top ten MCD-ranked voices, indicating that MCD may be a reasonable selector for potential top voices. By comparison, among voices selected for evaluation on MTurk, there was strong correlation of 0.8 to 0.9 between MTurk WER and Watson WER.

4.6 Utterance Selection for Low-Resource Languages: Amharic Audiobible

As one of the goals of this work is to develop methods for building voices out of found data for low-resource languages, it is important to determine which of our approaches actually generalize to other languages. Thus, we repeat some of our utterance-level training data selection experiments using Amharic data. We found online an audiobook of the Bible read by a male Amharic speaker in relatively good recording conditions. After preparing and formatting this data for TTS, we were able to train voices on subsets of this data and compare their intelligibility.

4.6.1 Audiobible Data

Audiobible data is a good potential source for found speech in a large variety of languages. In fact, the recently-released CMU Wilderness Multilingual Speech Dataset (Black, 2018) contains aligned speech and text from the New Testament in over 700 languages, collected online. We found a read audio version of the Bible in Amharic at `http://amharicniv.com`, and the text portion came from `http://www.bible.com`, both accessed on June 23, 2017. The data consists of about 55 hours of read speech from one male speaker. Each original audio file is an entire chapter, so the data was segmented by aligning it with the text using the Prosodylab-Aligner (Gorman *et al.*, 2011) and then splitting on punctuation.

For frontend processing we used Festival (Black *et al.*, 2014)[4]. The main things with which one needs to provide Festival to create a new language are a pronunciation lexicon and a phoneset

[4] Many thanks to Alan Black for providing us with some scripts and assistance in using Festival in languages other than the default US English.

definition. For the lexicon, we started with the Amharic lexicon from BABEL (Harper, 2011). There were many out-of-vocabulary (OOV) words that were present in the Bible text that this ASR lexicon did not contain, so we had to create pronunciations for these words and add them to the lexicon. We did this using the CMU Sphinx G2P tool[5]. We trained a g2p (grapheme-to-phoneme) model on the existing lexicon, and then used the model to generate phoneme sequences for our OOV words. The phoneset definition required a list of phonemes used in the lexicon, as well as indication of which ones are vowels. We created a custom questions file for our Amharic phoneme set; this can be found in Appendix C.

4.6.2 Experiments and Results

We used the Merlin "build your own voice" recipe, with a custom questions file that we created for our Amharic phoneset. Our baseline voice was trained on the entire 55 hours of data. Our test voices were four-hour subsets chosen on high, medium, and low values of the standard set of features we have been using: f0, energy, speaking rate, and level of articulation. Due to the difficulty of finding Amharic listeners on Amazon Mechanical Turk, we conducted an automatic evaluation for intelligibility by sending synthesized sentences to a speech recognizer trained by IBM for Amharic for the BABEL project[6]. WER results are in Table 4.11, with best five voices shown in bold.

While we would not necessarily expect high mean energy to be an indicator of good data to use for TTS according to the guidelines of (Matoušek *et al.*, 2008), it nevertheless continues to surface as one of our more consistently useful features across languages and corpora, being in the top 5 voices trained on both CALLHOME (Section 4.3.2) and MACROPHONE (Sections 4.4.1 and 4.5.2). Furthermore, we also did observe higher values for mean energy when we compared TTS data to other genres in Chapter 3 Section 3.5.1, indicating that this may in fact be a salient feature of TTS data. The other features are all what we might expect to be in line with standard TTS data: lower ranges for variation in energy and f0, as well as lower mean f0.

[5] `https://github.com/cmusphinx/g2p-seq2seq`

[6] Many thanks to Andrew Rosenberg for assistance with this evaluation.

Voice	IBM WER	Voice	IBM WER
Baseline	45.00%		
High Mean Energy	**20.00%**	High Mean F0	50.70%
Med Mean Energy	31.40%	Med Mean F0	50.70%
Low Mean Energy	64.30%	**Low Mean F0**	**26.40%**
High Stdv Energy	61.40%	High Stdv F0	38.60%
Med Stdv Energy	**30.70%**	Med Stdv F0	40.00%
Low Stdv Energy	**27.10%**	**Low Stdv F0**	**27.10%**
Slow Speaking Rate	62.90%	High Articulation	52.10%
Med Speaking Rate	41.40%	Med Articulation	40.00%
Fast Speaking Rate	45.70%	Low Articulation	41.40%

Table 4.11: ASR word error rates for voices trained on 4-hour subsets of Amharic audiobible data.

4.7 Conclusions

In these various experiments, we have seen that training voices on smaller amounts of data that have been chosen in a principled manner can in fact produce more intelligible voices, in particular when using read ASR data or found audiobook data. Guided by (Matoušek *et al.*, 2008), we selected training sets based on features that are related to the professional TTS style. Surprisingly, we found that training on utterances with slow speaking rate produced some of the *least* intelligible voices. Consistency in mean energy level also appears to be important for intelligibility, regardless of whether it is a high or low level – this may correspond with a low standard deviation of f0 in the *overall* training set. We found that when you have a large corpus of many speakers, it is a consistently better strategy to select training data at the speaker level rather than at the utterance level. Automatically identifying the most intelligible speakers with ASR and then training a voice just on the data from those speakers does produce a voice that is more intelligible to human listeners. Having a larger pool of data to choose from is also beneficial, even if you are choosing the same amount of data. We also found that the data selection approach generalizes to create more intelligible voices from single-speaker audiobible data in Amharic.

While Mechanical Turk was an effective tool for crowdsourcing our intelligibility evaluations, the turnaround time was very slow and we encountered difficulties finding speakers of our languages of interest other than English. We have found that both ASR and MCD correlate well with human evaluations for intelligibility, and are useful tools for both speeding up preliminary evaluations as well as evaluating voices in low-resource languages.

Chapter 5

Subset Selection for Naturalness

5.1 Introduction

Although data collected for training speech recognizers presents the challenge of training intelligible voices, we have found that other cleaner sources of data can be used to create voices that are intelligible but still present the challenge of naturalness. For example, even a very small amount of radio broadcast news can produce very intelligible voices, but since this data was still not collected specifically with TTS in mind, the voices lack naturalness and often have a choppy-sounding, uneven f0 contour. Thus, we focus primarily on ways to improve naturalness of voices trained on this type of data.

We have conducted a number of data selection experiments using voices trained on the Boston University Radio News Corpus (BURNC) data (Ostendorf *et al.*, 1995), which consists of about four and a half hours of speech from three professional female radio broadcast speakers, and a little over five hours of speech from four male speakers, recorded in clean conditions. We trained voices on fixed-size subsets based on acoustic and prosodic features, combinations of the best filtering methods, and removal of outliers. The pilot work presented in this chapter has been published in (Cooper *et al.*, 2016a) and (Cooper *et al.*, 2016b).

5.2 Related Work

There have been a number of studies using high-quality found data to build TTS voices that investigate the question of what portions of the data are best. (Gallardo-Antolín *et al.*, 2014) used radio broadcast news recordings to train synthetic voices, exploring different speaker diarization and noise detection techniques to remove unsuitable utterances automatically. Audiobooks have also been a popular source of found data for building TTS voies. In particular, (Chalamandaris *et al.*, 2014) used a corpus of audiobook speech to build unit selection voices. To handle the different recording conditions of the various audiobooks, they first performed a recording-condition-based clustering, and kept only utterances from one cluster. Since audiobook speech contains a great deal of variation and expressivity, such as emotion and character voices, they selected the most neutral utterances by plotting the mean and standard deviation of pitch, and keeping only the 90% of the data closest to the centroid. Furthermore, since the 140 hours of speech in their corpus had to be aligned with the text automatically, they were able to remove sentences with a low alignment score, in order to remove both poorly-aligned data as well as sentences where the speaker did not read the text exactly as written. They found that the combination of these approaches did produce a better voice. Similarly, (Watts *et al.*, 2013) built a corpus of 60 hours of speech from audiobooks in 14 languages, one speaker per language, also including only utterances with high automatic alignment confidence scores. They also created a module for selecting utterances with uniform speaking style (as opposed to more expressive utterances commonly found in audiobook speech) using a lightly-supervised active learning based approach, specifically for the purpose of building HMM-based voices in different languages. Finally, (Braunschweiler and Buchholz, 2011) also discarded low-confidence utterances, but based on ASR confidence rather than alignment confidence, and discarded utterances that were not neutral or suitable for a TTS corpus, as judged by a human. They also developed an automatic method for deciding utterance naturalness, based on discarding utterances outside of manually-chosen thresholds for acoustic features such as silences, utterance duration, f0, root mean square amplitude, and voicing, as well as text-based features such as punctuation and numbers which they deemed likely to result in text normalization errors. Despite discarding nearly half their original data, they found that the HMM voices they trained using both of these methods were judged as significantly better than using all of the data in a preference test, and the manual approach also did significantly better than the automatic one. These results all show promise for data selection methods on nontraditional but generally high-quality sources of TTS training data for producing better voices. Although these methods were developed primarily for audiobooks and for data from a single speaker, the general approach of data selection may also prove to be applicable to building parametric voices from other types of found data from multiple speakers.

Many of the features we chose to explore for data selection were guided by our lab's prior work on the acoustic features that correlate with charisma in American English, as well as in other languages such as Arabic and Swedish (Rosenberg and Hirschberg, 2005; Biadsy *et al.*, 2007; Biadsy *et al.*, 2008; Rosenberg and Hirschberg, 2008), which found that in American English, louder utterances, utterances higher in the speaker's pitch range, and utterances with a faster speaking rate were rated by listeners as more charismatic. Furthermore, high mean pitch and high standard deviation of root mean square intensity correlated with charisma cross-culturally. Since these standard and language-independent features are informative of charismatic speech, we also hypothesize that, as features that are clearly salient to listeners' perception of speech, they may play a role in perceived naturalness of synthesized speech as well.

5.3 Subsets Based on Acoustic and Prosodic Features

To build voices for our experiments, we created one-hour training data subsets of the BURNC corpus by selecting utterances based on a number of different criteria. Similar to our intelligibility experiments, we trained voices only on all-male or all-female data to produce more consistent models. We compared these subset voices to one of two baselines, one (for the female voices) trained on all of the female data (4h 40m) and one (for the male voices) trained on all of the male speech (5h 15m). For the baselines and our selected subsets, utterances were defined as sentences in the transcript text, and the audio was segmented accordingly.

We selected subsets of the male and female utterances based on a number of different criteria. These criteria are based upon factors we hypothesized might be useful for selecting training data sets to create more natural-sounding voices. We examined mean and standard deviation of energy and fundamental frequency (f0), selecting subsets of utterances totaling one hour in duration for the highest, lowest, and middle of each of these values, computed using the Praat speech analysis software (Boersma, 2001). We did the same for speaking rate, defined by syllables per second. We also considered that hypo- and hyper-articulation of training utterances might have an effect on the naturalness of the resultant voice, so we selected subsets of low- and high-articulation utterances, computed by mean energy divided by speaking rate, also each one hour. Unlike the ASR corpora, the BURNC data was recorded under professional conditions and is read professional speech, so our previous approaches of removing utterances with noise or disfluencies were not relevant for this data. We also hypothesized that there may be some optimal utterance length for TTS training data, so we selected subsets of the longest, shortest, and median utterances based on the length of the audio file for each utterance. The time ranges of these subsets can be found in Table 5.1.

Female		Male	
Subset	Range (s)	Subset	Range (s)
Short	0.25 - 5.50	Short	0.71 - 5.24
Middle	5.25 - 6.98	Middle	6.40 - 7.22
Long	10.05 - 22.19	Long	11.22 - 25.39

Table 5.1: Range of utterance durations for hour-long subsets of short, middle, and long utterances.

We trained our TTS voices using the Hidden Markov Model Based Speech Synthesis System (HTS) (Zen *et al.*, 2007). We based our training recipe for the baselines and for the data selection subsets described above on the speaker-independent training demo recipe. Although speaker-adaptive training (SAT) is known to produce more stable models and therefore better-sounding voices (Yamagishi, 2006), we found that with our data, there was no preference between SAT and SI voices trained on all of the male data, and there was a slight but not significant preference for the SAT voice over the SI voice trained on the female data. Given the greater computational resources required to train SAT voices, we decided to use SI training for this set of experiments.

We obtained the standard set of full-context phonetic labels for the BURNC data using the Festival Speech Synthesis System (Black *et al.*, 2014). For this set of experiments, we evaluated the naturalness of the female voices using a Mean Opinion Score (MOS) test. We later decided to switch to a much simpler pairwise comparison task for the male voices and for all subsequent experiments. MOS results for the female voices are in Table 5.2 and include ratings for the intentionally very robotic-sounding "Zarvox" voice from the Mac OS X `say` command, and a human voice speaking the same sentences, as attention checks and anchor points. Pairwise results for the male voices are in Table 5.3.

Voice	Rating	Voice	Rating
Robotic	1.03	Low mean energy	2.41
High mean f0	1.97	Mid mean energy	2.41
Hyper-articulated	2.08	Longest utts	2.5
High mean energy	2.08	Fast rate	2.55
Mid length utts	2.08	Mid mean f0	2.55
Slow rate	2.13	Mid sdev f0	2.6
High sdev energy	2.13	Low sdev f0	2.6
Mid sdev energy	2.28	Baseline	2.68
Shortest utts	2.33	Hypo-articulated	2.7
High sdev f0	2.37	Low mean f0	2.7
Low sdev energy	2.37	Natural speech	4.95
Mid rate	2.4		

Table 5.2: Average naturalness rating for each voice (low to high), MOS experiment

For the female voices, only voices trained on subsets of hypo-articulated utterances and low mean f0 utterances had a higher MOS than the baseline voice, but the difference was not significant. Indeed, we see very little difference in the scores for the various test voices, which is part of what motivated us to switch to the pairwise preference test. For the male voices, no subset voice was preferred over the baseline – we hypothesize that this is because the four male speakers sound more similar to each other than the three female speakers sound to each other, resulting in more consistent models and an overall better-sounding male baseline voice with little room for improvement.

5.4 Varying Subset Sizes

We took our two most promising features for female voices from our first set of experiments, hypo-articulated and low mean f0, and created 30-minute and 2-hour training data subsets of the utterances to complement our original 1-hour subsets. Results for pairwise comparisons for

Male Voice	Preferred	P-value
Low mean f0	16.7%	2.42e-7
Mid mean energy	20%	3.36e-6
Mid mean f0	20%	3.36e-6
Slow speaking rate	21.7%	1.14e-5
Hyper-articulated	23.3%	3.61e-5
Low sdev f0	25%	1.08e-4
Fast speaking rate	25%	1.08e-4
Medium length utts	25%	1.08e-4
High stdv energy	28.3%	7.89e-4
Mid stdv energy	28.3%	7.89e-4
High sdev f0	28.3%	7.89e-4
Mid articulated	28.3%	7.89e-4
Mid speaking rate	28.3%	7.89e-4
High mean energy	30%	1.95e-3
High mean f0	30%	1.95e-3
Middle sdev f0	30%	1.95e-3
Shortest utts	31.7%	4.51e-3
Low mean energy	35%	0.02
Hypo-articulated	35%	0.02
Longest utts	38.3%	0.07
Low stdv energy	41.7%	0.20

Table 5.3: Percent of votes for the test voice over the male baseline (out of 60 ratings; low to high), and p-value.

naturalness are presented in Table 5.4.

	Hypo-articulation		Low Mean F0	
Amount	Preferred	P-value	Preferred	P-value
30min	31.7%	4.51e-3	36.7%	0.04
1hr	43.3%	0.30	53.3%	0.61
2hr	58.3%	0.20	56.7%	0.30

Table 5.4: Pairwise comparison preferences for female voices trained on subset sizes of 30 minutes, 1 hour, and 2 hours over the baseline.

Over both features, we see steadily increasing preference for larger data sets, indicating that for this type of data and these selection methods, more data is better. However, none of these subsets produced voices rated to be significantly better than the baseline.

5.5 Combination of Best Approaches

We next hypothesized that some combination of our best approaches so far might produce a better voice. We tried a number of different combination methods of our top two features, hypo-articulation and low mean f0:

1. We took a 2-hour subset of the most hypo-articulated utterances and intersected it with a 2-hour subset of the lowest mean f0 utterances to produce a 54-minute training set of female data;

2. we combined 30-minute subsets of each by set union into a 56-minute minute subset (not a full hour because some utterances appear in both sets);

3. we combined 1-hour subsets of each into a 1 hour and 46 minute minute subset;

4. we multiplied the mean f0 values for every female utterance by their articulation values, and selected a 1-hour subset of the utterances with the lowest resulting values;

5. same as (4) except we selected a 2-hour subset;

6. same except a 3-hour subset;

7. same except a 4-hour subset.

Pairwise preference results are in Table 6.6, with results significantly better than the baseline in bold.

Combination	Preferred	P-value
1. Intersection (54min)	53.3%	0.61
2. Union (56min)	58.3%	0.20
3. Union (1hr46min)	61.7%	0.07
4. Multiplication (1hr)	61.7%	0.07
5. Multiplication (2hr)	**68.3%**	0.005
6. Multiplication (3hr)	51.7%	0.80
7. Multiplication (4hr)	45.0%	0.44

Table 5.5: Pairwise comparison preferences for female voices trained on different subsets of combinations of hypo-articulation and low mean F0.

We see that subsets (3) and (4) perform well, approaching significance, with (5) performing significantly better than the baseline ($p \leq 0.05$), indicating that combining our best filtering methods can improve naturalness. This motivated us to try (6) and (7), using the same filtering method but with increasingly more data, however more data in this case did *not* turn out to be better – the optimal subset size selected using this approach was 2 hours.

5.6 Removal of Outliers

We find that with this type of data, which is relatively high-quality, removing too much of it and using small subsets is generally detrimental. So, we trained some voices by removing a smaller portion of the data – outlier utterances based on the features that produced the *worst* voices in our prior experiments. Since we have seen so far that utterances with speaking rates at the extremes and hyper-articulated utterances produced some of the worst voices, we created sets where we trimmed the upper and lower tail of the female utterances when sorted by speaking rate (two separate sets), and the upper tail of hyper-articulation. For high speaking rate, we found that the mean was 4.66 syllables per second and the standard deviation was 0.75, so we chose a cutoff of mean plus one standard deviation (5.40) which would then include 88.49% of the total data, or 3 hours and 52 minutes. For removing low speaking rate utterance outliers, we did a similar cutoff of mean minus one standard deviation, giving us a subset of 4 hours and 6 minutes. As for hyper-articulation, where articulation was computed as mean energy divided by speaking rate, looking at female data only, we found a mean of 13.91 and a standard deviation of 2.67; we again chose a cutoff of mean plus 1 standard deviation (16.57). This gave us 93.86% of the original dataset, or 4 hours and 6 minutes of data remaining. Pairwise comparison results for these two voices are in Table 5.6. We have here obtained a significantly more preferred voice ($p \leq 0.05$) by removing the outlying hyper-articulated utterances from the training set.

Outlier feature	Preferred	P-value
High speaking rate	56.7%	0.30
Low speaking rate	51.7%	0.80
Hyper-articulation	**65.0%**	0.02

Table 5.6: Pairwise comparison preferences for female voices trained on data sets with outliers removed.

5.7 Conclusions

Level of articulation has shown to be a consistently useful feature for all of our approaches, especially in the case of removing hyper-articulated outlier utterances and when combining hypo-articulation with low mean f0, both of which produce female voices that were rated as sounding significantly more natural than the baseline. This indicates that radio broadcast news is slightly more expressive than ideal data for TTS, but choosing the more neutral utterances can remedy this and improve naturalness. The two sides of our approach, identifying the best utterances to train on, as well as identifying outlier utterances that are best left out of the training data, have both proven successful. We have also seen that combining our best individual selection features was useful for improving naturalness.

Chapter 6

Adaptation, New Frontend Features, and Other Modeling Approaches

6.1 Introduction

We have observed that with relatively high-quality data, such as BURNC (Ostendorf *et al.*, 1995) radio broadcast news, keeping more of the data may be generally preferable to using smaller subsets. There is not much we want to exclude, given the absence of noise, disfluencies, and irregular speech. Nevertheless, we may have something to learn from the features that we have been investigating as subset selectors. This section presents a variety of approaches for training voices using all of the data, while also making use of the knowledge of our acoustic and prosodic features to modify how the voices are modeled to improve naturalness. We also explore the question of how much high-quality data is needed, and whether combining high- and low-quality data can produce any benefits in terms of intelligibility.

6.2 Related Work

There have been a variety of studies exploring the use of adaptation in TTS for the purpose of handling noisy, heterogeneous data. For instance, in (Yamagishi *et al.*, 2008), noisy recordings of political speeches were used to adapt an average HMM voice trained on clean data from many speakers. The authors obtained a robust, natural-sounding voice with performance minimally degraded by the inclusion of noisy data. They also discovered that, by using recording-condition-adaptive training, they could produce more stable synthetic speech. (Karhila *et al.*, 2013) also trained AVMs on clean data from a TTS corpus and adapted it to a target speaker using data with added noise. They found that listeners could in fact distinguish between voices adapted with clean and noisy data, but that naturalness and speaker similarity were not affected. (Dall *et al.*, 2012) used human judgments of perceptually-similar source speakers, as well as objective measures, for building an AVM to create a better adapted voice than one based on an AVM trained on all of the source speakers. Similarly, (Govender and de Wet, 2016) aimed to select the best adult source speakers for building an AVM to adapt to child speech based on the objective measures of MCD and root mean squared error of log f0, finding that these measures also correlated well with human judgments of intelligibility for the adapted voices. (Yamagishi *et al.*, 2010) trained an AVM on data collected in an office environment and adapted to cleanly-recorded speech. They found that using both noisy and clean data together produced a voice with a slightly (but not statistically-significantly) higher mean opinion score than a voice trained on clean data alone, and concluded that more data, even of a lesser quality, can be beneficial. They also explored the question of how much target speaker data is needed to train a speaker-dependent voice on that speaker's data alone as opposed to using it for adaptation in combination with an AVM. They found that it is better to use the data for adaptation when less than an hour of target speaker data is available, and it is better to train a speaker-specific voice on the target speaker data alone if there is more than two hours available. We explore similar questions of quantity and quality in our experiments in Section 6.7.

There has also been some interesting work on the use of novel frontend features to create different types of voices. (Yamagishi *et al.*, 2003) used an approach called "style-mixed modeling" to train a voice that could speak in different styles. This entailed using data from one speaker in different styles ("reading," "rough", "joyful," and "sad"), labeling those styles at the frontend along with the standard set of linguistic features, and then choosing which style to synthesize, by including the chosen style label at the frontend. More recently, (Henter *et al.*, 2017) learned emotional nuance from a database of emotional speech, and enabled structured control of these acoustic characteristics by adding corresponding features at the frontend. In our own frontend-labeling experiments, rather than using categorical speaking styles or emotions, we are using measurable acoustic and prosodic features and treating ranges of each of these as their own “manner of speaking,” aiming for speech that is close to more typical TTS data, in order to produce the most natural synthesis.

6.3 Interpolation

We noticed informally that many of the HMM-based voices we produced using BURNC data had a very choppy-sounding f0 contour, despite our data selection. We hypothesized that a simple way to remedy this would be to set the f0 values to a constant in the training data, train a monotone voice with that training data, and then interpolate the monotone voice with the baseline voice in equal proportions. This effectively halves the variance of the generated log-f0 sequence. We therefore created a female monotone voice, from which we were able to create an interpolated voice using the baseline and the monotone models. We set the interpolation weights to be equal. The female interpolated voice was preferred by 63.3% (significant at p=0.04) (Cooper *et al.*, 2016b), a surprising improvement for such a simple approach, which validates our hypothesis that the erratic, choppy f0 contour does in fact detract from naturalness.

6.4 Subset Adaptation

Our results so far indicate that limiting the size of the training data for high-quality data such as BURNC may not be beneficial. With this in mind, we decided to try training voices on *all* of the female data, indicating which subset each utterance belongs to, running adaptive training, and adapting the average voice to each of our subsets, one voice per subset. Adaptive training was conducted using the HTS speaker-adaptive training recipe (described in detail in Chapter 2 Section 2.2), except instead of adapting to speakers, we adapted to the subsets. By treating the subset (e.g. “high mean f0”) as one “speaker” and the rest of the data as another “speaker,” we hope to obtain the benefits both of using *all* of the data and also of identifying features of utterances that may be more ideal for producing a natural voice. We used this approach on each of the 1-hour subsets we used in our prior work for the female data (Chapter 5, Section 5.3): we explored the same features related to manner of speaking, namely mean and standard deviation of f0 and energy, identified automatically using Praat (Boersma, 2001); speaking rate in syllables per second; level of articulation, defined as mean energy divided by speaking rate; and duration of the utterance.

1-hour subsets were the hour of high, medium, or low-valued utterances for each of these features. Each adapted voice was compared in a pairwise manner to the baseline voice trained on the same data without adaptation. Results are shown in Table 6.1, and have been published in (Cooper *et al.*, 2016a).

Our two best approaches here were the voices adapted to 1-hour subsets of hypo-articulated utterances and middle mean energy utterances for female data, with hypo-articulation approaching significance, but none significantly better than baseline. As with our speaker-independently trained subset voices, we examined additional variations of 30-minute and 2-hour subsets for subset adaptation based on these two features. We also tried combining these two best approaches by intersecting 2-hour sets of each to produce a 59-minute subset. Results are presented in Table 6.2. Again, each voice is compared to the baseline trained on all of the female BURNC data without adaptation.

Adapting to subsets of 30 minutes and 2 hours does worse than adapting to 1-hour subsets, for both selection approaches of hypo-articulation and middle mean energy, indicating that perhaps 1-hour adaptation sets are best to use. Intersecting 2-hour sets of our best approaches to get an approximately 1-hour set for adaptation did not turn out to be useful.

6.5 Adaptation vs. Labeling Acoustic Properties at the frontend

In this section, we expand upon our earlier subset adaptation experiments, and we also introduce another approach that makes use of all the data from a corpus: rather than treating ranges of these feature values as regression classes and adaptively training, we add in the ranges for these feature values at the frontend. Each utterance is labeled as "high," "middle," or "low" for a given feature, such that the data is divided into thirds. Then, we synthesize our test utterances using each of those three settings. We aim to identify which settings produce the most natural synthetic speech.

One limitation of our initial adaptation experiments described earlier in Section 6.4 was the unequal and split nature of the regression classes. When adapting to a "middle range" subset, the "not-in-subset" class is comprised of two disparate parts of the data — low- and high-valued utterances for that feature. Combining these may have harmed the model's consistency. Thus, in this work, we explore the use of three classes, "high," "middle," and "low," each consisting of one

Adaptation Subset	Preferred	P-value
High mean energy	35.0%	0.02
High mean F0	40.0%	0.12
Hyper-articulation	41.7%	0.20
Low mean energy	43.3%	0.30
Middle mean F0	43.3%	0.30
Slow speaking rate	45.0%	0.44
High std.dev energy	46.7%	0.61
Middle std.dev F0	46.7%	0.61
Short duration	48.3%	0.80
High std.dev F0	48.3%	0.80
Fast speaking rate	50.0%	1.0
Low mean F0	50.0%	1.0
Medium duration	50.0%	1.0
Low std.dev energy	50.0%	1.0
Long duration	51.7%	0.80
Middle std.dev energy	53.3%	0.61
Medium speaking rate	56.7%	0.30
Low std.dev F0	56.7%	0.30
Middle mean energy	60.0%	0.12
Hypo-articulation	61.7%	0.07

Table 6.1: Pairwise preferences for female voices adapted to 1-hour subsets.

Adaptation Subset	Preferred	P-value
Hypo-articulation - 30min	56.7%	0.30
Hypo-articulation - 2hr	48.3%	0.80
Middle mean energy - 30min	50.0%	1.0
Middle mean energy - 2hr	53.3%	0.61
Intersection - 59min	48.3%	0.80

Table 6.2: Pairwise preferences for female voices adapted to 30-minute, 2-hour, and intersected sets of hypo-articulated and middle mean energy utterances.

third of the data, rather than adapting to hour-long subsets. We hypothesize that this approach will produce more consistent models with better ability to adapt towards the desired speaking characteristic. Adapting to subsets of one third of the data, rather than subsets of one hour each, will also enable a more direct comparison to the frontend labeling approach, which uses the same partitions of thirds of the data. This work has been published in (Cooper and Hirschberg, 2018).

6.5.1 Tools and Corpora

We initially trained our TTS voices using the Hidden Markov Model Based Speech Synthesis System (HTS) (Zen *et al.*, 2007), version 2.3, using the hts_engine vocoder. However, with recent advances in neural network based speech synthesis, we also wished to learn which results generalize across acoustic model types, so we repeated these adaptation and frontend labeling experiments using the Merlin (Wu *et al.*, 2016) toolkit for neural network based voice training, with the WORLD (Morise *et al.*, 2016) vocoder. For text processing, we used the default U.S. English frontend for Festival (Black *et al.*, 2014). We are again using the BURNC corpus of professional radio broadcast news from three female speakers. We evaluated all of our voices for naturalness using our pairwise naturalness comparison task on Amazon Mechanical Turk.

As in our earlier experiments, we explored features related to manner of speaking, namely mean and standard deviation of f0 and energy, identified automatically using Praat (Boersma, 2001); speaking rate in syllables per second; level of articulation, defined as mean energy divided by speaking rate; and duration of the utterance. For each of these features, we sorted our training utterances by feature value and then divided the data into thirds, labeling each utterance as having

Feature	hi	med	lo
Mean f0	40.0	53.3	**56.7**
Std. dev f0	33.3	38.3	**43.3**
Mean energy	41.7	**60.0**	58.3
Std. dev energy	**43.3**	41.7	40.0
Speaking rate	**46.7**	**46.7**	35.0
Articulation	38.3	30.0	**40.0**
Duration	**40.0**	31.7	36.7

Table 6.3: Percent preference for HTS voices trained adaptively using high, middle, and low partitions for each feature.

a high, medium, or low value for that feature as appropriate. For the adaptation approach, we treated each third of the data in turn as an adaptation set and used it to adapt an average voice model (AVM). For the frontend labeling approach, we introduced a new contextual feature, added on to the standard set of contextual features for English. This new frontend feature took on the value of high, middle, or low as appropriate for each utterance. We also added our new contextual features to the test output labels, creating high, medium, and low-setting label files, in order to compare synthesis output at each setting. We compare each voice to a baseline trained on the same data (all female BURNC utterances), but with no adaptation or extra frontend features.

6.5.2 HMM-Based Synthesis Experiments

6.5.2.1 Adapted Voices

We adaptively trained one voice per feature, using all of the female BURNC data, and then synthesized test utterances adapted to each of the three classes (high, medium, and low values for the given feature). To accomplish this, we used the HTS speaker-adaptive training recipe, but, instead of labeling different speakers, we labeled utterances as high, middle, or low. Results are shown in Table 6.3, with best settings for each feature in bold. Preferences are in comparison with a voice trained on the same data but with no adaptation.

While low mean f0 and middle mean energy adapted voices were rated as better than the baseline, neither of these preferences turned out to be statistically significant.

6.5.2.2 Contextual Feature Labeled Voices

Another way to make use of all of the data while also making use of informative acoustic and prosodic features is to label each utterance as having a high, middle, or low value for a given feature at the *frontend*, as part of the set of contextual features. One major benefit of this approach is that, in the construction of the decision trees for Hidden Markov Model based synthesis, if there are any contextual features that are not actually informative in splitting the data, they simply will not be used. Therefore, we are able to add arbitrarily many new contextual features, which, if they do not contribute to better modeling of the data, simply will not appear in the decision trees.

The standard set of contextual features is obtained using Festival, and includes phoneme-level information such as the previous two, current, and next two phonemes; the position of the current phoneme in the syllable; position of the current syllable in the word; whether the syllable is stressed or not; position of the current word in the phrase; and similar features providing a linguistic representation. The full list of standard contextual features for US English can be seen in Appendix A. Using the same partitions of the data into thirds, we added one new contextual feature to our full-context labels, indicating whether the utterance has a high, middle, or low value for one particular feature; we also added relevant questions to the HTS *questions file*. The questions file for HTS voice training contains a variety of yes/no questions that are used in the construction of the acoustic model decision tree, each followed by patterns for which a match in the full-context label would indicate a "yes". We added three new questions that ask whether the new feature value is high, medium, or low. We then trained one voice for each feature on all of the data labeled as described, and then synthesized test utterances with each of the three settings. Results are shown in Table 6.4, with the best setting out of high, medium, or low in bold, and statistically-significant preferences underlined. Each voice is rated in comparison to the baseline voice trained on the same data but with no additional contextual features added.

Synthesizing with the low setting for standard deviation of f0 and with the high setting for duration both produced speech that was significantly preferred over the baseline. The success of the low standard deviation of f0 setting makes sense because professional speakers for a TTS corpus are typically instructed to speak with as little variation as possible (Matoušek *et al.*, 2008). For the "high duration" synthesis, we are not necessarily synthesizing long utterances, but rather choosing to synthesize *in the style of* the longer utterances in the training data. This may have resulted

Feature	hi	med	lo
Mean f0	55.0	**60.0**	51.7
Std. dev f0	60.0	55.0	**63.3**
Mean energy	48.3	**56.7**	45.0
Std. dev energy	**51.7**	50.0	**51.7**
Speaking rate	**50.0**	46.7	45.0
Articulation	**56.7**	**56.7**	**56.7**
Duration	**63.3**	50.0	56.7

Table 6.4: Percent preference for HTS voices trained with labels for high, medium, or low values for acoustic and prosodic features and then synthesized at each of the three settings.

in better naturalness ratings because longer training utterances provide more speech in a natural context.

When adding new contextual features and questions about them for decision tree construction, it is useful to know how informative these new features actually are. Questions are chosen at each node split to maximize the likelihood of the data under the model. Questions that are higher up in the tree are therefore more informative, and some questions which are not informative may not end up in the decision tree at all. Questions may also appear in multiple places in the tree. In HTS voice training, there are separate decision trees for f0, spectrum, and duration, and furthermore, there are separate trees constructed for each stream, e.g. delta and delta-delta. For simplicity, due to multiple trees and possibly multiple instances of our questions in each tree, we report in Table 6.5 the *minimum* depth of each question related to our new feature for each voice, over all of the trees for that voice. Lower values imply that the feature was more informative in splitting the data.

Interestingly, informativeness of the features does not consistently correspond to preference for the voice using that feature. Minimum depth over all trees is a somewhat coarse measure; it may be interesting in future work to investigate this in a more detailed way, e.g. to see whether minimum depth in *certain* trees corresponds better with preference for the voice.

Next, we wanted to see whether combining features could produce even more improvement. Rather than trying all combinations and all settings of the features, we accumulated features one by one in the order that they gave improvement, and only synthesized using the best setting for

Feature	hi?	med?	lo?
Mean f0	4	6	2
Std. dev f0	6	7	6
Mean energy	2	4	2
Std. dev energy	3	3	2
Speaking rate	2	4	5
Articulation	2	6	5
Duration	5	8	5

Table 6.5: Minimum depth of each question over all decision trees for each voice.

each feature.

For some features it was not clear which setting was "best" – in particular articulation and standard deviation of energy. So we posted tiebreaker tasks on MTurk. The tie was not resolved for articulation, so we picked the low setting, corresponding with our prior findings (Cooper *et al.*, 2016b; Cooper *et al.*, 2016a) that training on hypo-articulated utterances tends to produce better voices. For standard deviation of energy, the low setting was slightly preferred. Comparisons are made to a baseline voice trained on the same data but with no additional contextual features added.

Synthesizing with only the best setting for each feature, our features gave improvements over the baseline in the following order, from most to least: duration (hi), standard deviation of f0 (lo), mean f0 (med), articulation (lo), mean energy (med), standard deviation of energy(lo), and speaking rate (hi). We thus trained six new voices: the first, with both duration and standard deviation of f0 labeled in the contextual features and with the "hi" and "lo" settings for those features, respectively, chosen at synthesis; the next voice, with those same features plus mean f0, set to the "med" setting at synthesis; and so on. Preferences over the baseline are shown in Table 6.6; note that each line of the table represents features from the preceding line *plus* the new feature added on the current line.

Surprisingly, the best two features, which on their own resulted in better voices (Table 6.4), produced a worse voice in combination. We see improvements as we add each feature, with the exception of adding speaking rate, which results in a slight drop in naturalness ratings. These features appear to be interacting in unexpected ways.

Features	Preference
Duration (hi) + Std. dev. f0 (lo)	46.7
+ Mean f0 (med)	53.3
+ Articulation (lo)	56.7
+ Mean energy (med)	58.3
+ Std. dev. energy (lo)	65.0
+ Speaking rate (hi)	60.0

Table 6.6: Percent preference for HTS voices trained with labels for multiple features

6.5.3 Neural Network Synthesis Experiments

Neural network based synthesis has recently produced very high-quality voices, and addresses some of the naturalness issues common to HMM-based voices. Thus, we explored neural network-based voice training in addition to HTS in order to determine experimentally whether these advances generalize to the type of data we are using.

We repeated our experiments using the Merlin toolkit for neural network based synthesis (Wu *et al.*, 2016). For the baseline and frontend feature experiments, we used the basic "build your own voice" recipe from Merlin, using WORLD for feature extraction and vocoding.

First, we trained a baseline voice using this recipe with all of the female BURNC data using the standard full-context labels extracted by Festival. When we compared this to the HTS baseline using the same audio and labels, the preference for the Merlin voice was **90.0%**. It is therefore apparent that not only does neural network based synthesis produce more natural voices when trained on standard TTS data, but on radio broadcast news data from multiple speakers as well.

6.5.3.1 Adapted Voices

For the adaptation experiments, we trained an AVM on all of the female data and then adapted to each subset using the Merlin speaker adaptation recipe, which implements two different types of adaptation (described in (Bollepalli *et al.*, 2017)): back-propagating the adaptation data through the model to re-tune all the weights ('fine-tune'); and "Learn Hidden Unit Contributions" (LHUC), which recombines hidden units based on the adaptation data (Swietojanski *et al.*, 2016). We tried both methods, and we found that the best voices were produced using the 'fine-tune' adapta-

Feature	hi	med	lo
Mean f0	43.3	**45.0**	36.7
Std. dev f0	48.3	**60.0**	50.0
Mean energy	**53.3**	45.0	36.7
Std. dev energy	36.7	**43.3**	36.7
Speaking rate	**45.0**	**45.0**	41.7
Articulation	**50.0**	45.0	45.0
Duration	41.7	45.0	**60.0**

Table 6.7: Percent preference for Merlin AVM adapted to subsets of the data selected based on high, middle, or low values for various acoustic and prosodic features.

tion method; full results for fine-tune adapted voices are shown in Table 6.7. Again, preferences are evaluated against the baseline voice trained on all of the female BURNC data, but with no adaptation.

Adapting to short duration utterances and adapting to middle standard deviation of f0 were both preferred over the baseline by 60%, which was not statistically significant.

6.5.3.2 Contextual Feature Labeled Voices

We repeated our experiments from Section 4.3, adding one new feature at the frontend that takes on a value of high, medium, or low depending on the utterance's value for the given acoustic or prosodic feature we are measuring, and then synthesizing output utterances with high, medium, and low settings for that feature. Neural-network-based synthesis differs from HMM-based synthesis in that neural-network-based synthesis does not make use of decision trees. Instead, the frontend features are converted into a binary sequence by way of the questions file, corresponding to "yes" and "no" answers for each question. Pairwise preference results for Merlin subset voices versus the baseline are presented in Table 6.8, with the **best** setting (out of high, medium, or low) for each feature in bold, and results significantly better than the baseline underlined. Again, the baseline is a voice trained on all of the female BURNC data but with no additional contextual features added.

We see that a number of voices are rated as more natural than the baseline (more than 50%), with one significant preference: the voice with mean f0 level labeled at the frontend, and test

Feature	hi	med	lo
Mean f0	41.7	53.3	<u>**65.0**</u>
Std. dev f0	51.7	**55.0**	50.0
Mean energy	46.7	48.3	**55.0**
Std. dev energy	**61.7**	50.0	60.0
Speaking rate	**50.0**	41.7	48.3
Articulation	41.7	41.7	**53.3**
Duration	48.3	**55.0**	50.0

Table 6.8: Percent preference for Merlin voices trained on data labeled as having high, medium, or low values for features and then synthesized with each of the three settings.

utterances synthesized with the “lo” setting, which is in fact consistent with low mean f0 being a useful feature for intelligibility in our data selection experiments in Chapter 4, as well as with the expectations for standard TTS data according to (Matoušek *et al.*, 2008). Although our best Merlin voice is not produced using the same features as our best HTS voice, and although the best settings for each feature are not the same across training methods, we do observe that this frontend-labeling approach can produce significantly more natural voices regardless of the acoustic model.

Next, we wished to see whether the combination of features with their best settings could lead to greater improvement, as we tried with HTS. We added features one at a time and trained voices using them, and synthesized using the *best* setting for those features as indicated in bold in Table 6.8. Since we had a three-way tie between medium standard deviation of f0, low mean energy, and medium duration, we posted tiebreaker tasks on MTurk to decide the order in which to add those features. Results for voices with accumulated features are in Table 6.9, and again the baseline for comparison is a voice trained on the same data but with no additional contextual features added.

We see again that combining the best two features does actually not do as well as using each feature separately, and in fact, this time, we generally see a *decrease* in naturalness ratings in comparison with the baseline as we add more features. It is possible that this is a result of overfitting from adding too many new features, or possibly that the different features are interacting in ways that hurt naturalness. While adding features in order from biggest to least improvement

Features	Preference
Mean f0 (lo) + Std. dev. energy (hi)	53.3%
+ Duration (med)	48.3%
+ Mean energy (lo)	46.7%
+ Std. dev. f0 (med)	56.7%
+ Articulation (lo)	35.0%
+ Speaking rate (hi)	46.7%

Table 6.9: Percent preference for Merlin voices trained with labels for multiple features combined

seemed sensible, perhaps experimenting with adding them in different combinations would be a possibility for future experimentation.

6.5.4 Conclusions

For these neural network based experiments, we computed inter-annotator agreement using Fleiss' Kappa, which we found to be 0.014. This a low value for agreement, indicating that listener preferences do vary considerably. The low agreement is also not surprising in light of the fact that very few of these voices were overall significantly preferred to the baseline.

We have found that, for both HMM-based synthesis and neural network based synthesis, adding *individual* acoustic and prosodic features as new frontend labels can significantly improve voice naturalness, but that combination generally does not help. We also have found that subset adaptation was generally not successful; however perhaps combining the two approaches may be a direction for future work. Furthermore, given the success of the frontend-labeling approach, using actual numerical values rather than discretized high, medium, and low settings may be a promising direction as well, since neural network synthesis allows for this type of input, unlike HMM-based synthesis.

6.6 Frontend Features for Turkish

In an effort to determine the extent to which our frontend-labeling results generalize to other languages, we repeated the experiments using data from the Turkish Broadcast News corpus (Saraçlar, 2012). We used a clean subset of the data consisting of four and a half hours of speech from three female news anchors, to be comparable with the similar amount of data that we have from female BURNC speakers. We manually picked utterances for this set, excluding utterances with poor recording quality, background music or noise, and utterances whose transcripts include items marked as unintelligible or hesitations.

Frontend processing for this data was done by starting with the BABEL (Harper, 2011) pronunciation lexicon for Turkish. Pronunciations for out-of-vocabulary words (words contained in the broadcast news transcripts but not in the BABEL lexicon) were obtained by training a pronunciation model on the original lexicon using Sequitur G2P (Bisani and Ney, 2008) and then using the trained model to generate phonetic sequences for the missing words. Phonetic alignments were then obtained using the EHMM (Ergodic Hidden Markov Model, (Prahallad *et al.*, 2006)) alignment module in Festival (Black *et al.*, 2014). We created a custom questions file for the BABEL Turkish phoneme set, which can be seen in Appendix D.

We added individual new frontend features in the same way that we did for the English BURNC experiments. Features were the same – low, middle, and high levels of mean and standard deviation of f0 and energy (extracted with Praat (Boersma, 2001)), speaking rate in syllables per second obtained from the Festival alignments, duration of utterance, and level of articulation. While in our original English experiments we defined articulation level as mean energy divided by speaking rate, we later considered that high articulation levels may also include a large variation in f0, so we also defined another measure of articulation which was computed as mean energy, divided by speaking rate, multiplied by standard deviation of f0. The original articulation measure that combines two features is denoted as Articulation (2), and the new articulation measure that combines three features is named Articulation (3).

We trained voices using the default "build your own voice" recipe in Merlin. We created a custom questions file for the Turkish phoneset that also includes appropriate questions for high, medium, and low levels of our added features.

We evaluated the trained voices for naturalness using our pairwise comparison task on Mechanical Turk. We synthesized 12 sentences from each trained voice model. The text for the sentences came from portions of the broadcast news transcripts which were not used in training. We required listeners to be native speakers of Turkish. Each task consists of a pair of audio files which are the same sentence spoken by two different voices, the test voice and the baseline voice (trained on the

Feature	hi	med	lo
Mean f0	51.7%	53.3%	**65.0%**
Std. dev f0	58.3%	46.7%	53.3%
Mean energy	**68.3%**	55.0%	55.0%
Std. dev energy	61.7%	**65.0%**	41.7%
Speaking rate	48.3%	56.7%	48.3%
Articulation (2)	46.7%	55.0%	53.3%
Articulation (3)	**70.0%**	58.3%	55.0%
Duration	20.0%	48.3%	56.7%

Table 6.10: Percent preference for Merlin voices trained on Turkish broadcast news data labeled as having high, medium, or low values for features and then synthesized with each of the three settings.

same data but with no new experimental contextual features added). Percent preferences for each voice over the baseline voice are presented in Table 6.10, with significant preferences in bold.

We see that we can get improvements in naturalness with this frontend-labeling approach in Turkish as well. Low mean f0 stands out since there was also a significant preference for that feature in English (Table 6.8). High mean energy is another feature that stands out throughout multiple experiments: in addition to producing a more natural voice in this experiment, it was also an informative feature for subset selection for intelligibility for both Amharic and English in Chapter 4. We computed inter-annotator agreement using Fleiss' Kappa, finding a very low value of 0.004. This shows that listener preferences vary widely, as they did for English. The preference for synthesizing at the high setting for articulation was surprising, since the low setting was preferred for English. This may be due to language differences between Turkish and English. (de Jong *et al.*, 2015) found that native speakers of English produce longer syllables in English than native Turkish speakers speaking Turkish; Turkish speakers were also found to produce fewer pauses and repetitions than English speakers. While they did not explore differences in energy or f0 characteristics, the existence of language differences especially in speaking rate does allow for the possibility that different speaking styles are expected across languages, and thus different types of voices are preferred. Nevertheless it is still surprising that the higher level of articulation, which incorporates slow speaking rate, was preferred in Turkish, which has a typically faster speaking rate than English. Since there was no strong preference for voices that incorporated the speaking rate feature by itself, it may be that the other aspects of hyper-articulation (higher energy, more variation in f0) are preferred. Another explanation may be the effect of *speaker* differences. There are only three speakers for each language, and the speakers themselves have enough difference in speaking style that adjusting the voice towards a particular feature makes the voice sound more like one speaker. Indeed, we informally noticed that in both languages, certain voices sounded more like certain news anchors from the original data – perhaps listeners preferred voices that sounded more like certain speakers.

6.7 Quality Adaptation for Intelligibility

One research question of interest in the case of low-resource TTS is, if only a very small amount of high-quality data is available, is it better to use that data by itself, or to combine it with a larger amount of lower-quality data? At what quantity threshold of high-quality data does the answer to this question change? We attempt to answer this by combining a large quantity of lower-quality, mixed-speaker data collected for ASR, the MACROPHONE corpus (Bernstein *et al.*, 1994), with varying small amounts of high-quality, single-speaker data from BURNC (Ostendorf *et al.*, 1995), radio broadcast news which may be considered "found" data, and ARCTIC (Kominek and Black, 2003), which was specifically created for TTS.

We trained voices using the "build your own voice" recipe in Merlin (Wu *et al.*, 2016) using 5, 10, 20, and 40 minutes of data from each of the high-quality sources (BURNC speaker "f1a" or ARCTIC speaker "slt"), and a baseline voice using 10 hours from MACROPHONE. Then, we used the "speaker adaptation" Merlin recipe to train combined voices – the 10 hours of MACROPHONE data combined with one of the small sets of high-quality data, and then adapted, using the fine-tune adaptation method, towards the high-quality portion. We then synthesized semantically-unpredictable sentences using each of these voices and evaluated them for intelligibility using our transcription task in a Latin-square configuration on Amazon Mechanical Turk. We also included a natural human speech version of each sentence to obtain a top-line for intelligibility of these sentences. Word error rates are reported in Table 6.11. We also report sentence error rate (SER),

Voice	**WER**	**SER**	**Voice**	**WER**	**SER**
burnc 5	68.1%	96.7%	slt 5	56.7%	95.6%
burnc 10	70.0%	97.8%	slt 10	48.2%	94.5%
burnc 20	34.2%	85.7%	slt 20	37.2%	83.5%
burnc 40	27.0%	73.6%	slt 40	34.4%	85.7%
macrophone + burnc 5	39.6%	90.1%	macrophone + slt 5	41.4%	93.4%
macrophone + burnc 10	33.1%	87.9%	macrophone + slt 10	36.3%	89.0%
macrophone + burnc 20	28.6%	78.0%	macrophone + slt 20	32.0%	75.8%
macrophone + burnc 40	21.8%	80.2%	macrophone + slt 40	29.0%	76.9%
macrophone only	62.0%	95.6%	human	7.38%	41.8%

Table 6.11: Transcription word and sentence error rates for voices trained on one type of data as well as voices trained and adapted with combined low- and high-quality data

which is the percent of transcribed sentences which contain at least one error.

We observe that *all* of the combined-data adapted voices are more intelligible in terms of WER than each of their respective counterparts using only the high-quality data by itself. These differences are statistically significant according to a Welch's two-sided *t*-test, for high-quality data amounts of 5 and 10 minutes. For 20 and 40 minutes, the combined-data adapted voices are also rated as more intelligible than the voices trained on the same high-quality data by itself, but these differences are not statistically significant.

We also observe that all but three of the voices are significantly more intelligible in terms of WER than the baseline of just 10 hours of MACROPHONE data by itself, indicating that collecting even a small amount of high-quality data can be a very effective way to improve a voice. The exceptions are the 5 minutes of ARCTIC "slt" data by itself, which was more intelligible than the baseline but not significantly so, and the 5 and 10 minute sets of BURNC "f1a" data by themselves, which were rated as less intelligible than the baseline but also not significantly so. It also appears that the 20- and 40-minute BURNC voices are more intelligible than the respective ARCTIC voices trained on the same amounts of data, which is interesting because ARCTIC was collected specifically for TTS whereas BURNC was not; however, these differences are also not statistically significant.

Sentence error rate can give a measure of how useful the voices really are. We can see that

SERs are very high, with most transcriptions of all synthesized sentences containing at least one error. Nevertheless, it can also give us a sense of how difficult our task is – SER for the human voice speaking the set of SUS is also surprisingly high at 41.8%, indicating that the sentences themselves are very difficult, as many of the sentences contain words that are easily misheard or confusable for other words.

6.8 Conclusions

We have experimented with a variety of voice modeling and training approaches that make use of all of the training data, while also making use of what we know about standard TTS speech and the acoustic and prosodic features of the data that we have. Following our observation that voices trained on BURNC data often have a choppy-sounding f0 contour, our experiment building a voice that interpolates the choppy voice with a monotone one has validated our hypothesis that the choppiness hurts naturalness ratings. Building off of the subset selection experiments in the previous chapter, we attempted to train voices on all of the data and adapt to the subsets rather than only train on the subsets; however, this approach did not show much promise. However, using these feature-based subsets as new categorical frontend features instead of adaptation classes did work well, for both HMM and neural network based acoustic models, and for both English and Turkish. While combining these frontend features gave some small improvements in the HMM based voices, this was not the case for neural network voices, which raises questions about how these features interact and how best to combine them. Finally, we found that although subset adaptation did not work, quality adaptation using small amounts of high-quality data in combination with lower-quality ASR data gives large gains in intelligibility. We can conclude from this that even a small amount of high-quality data can be beneficial, and that combining quantity and quality is better than using either one alone. We can also conclude that the larger amounts of noisy data can provide intelligibility gains over just using a small amount of high-quality data by itself.

Chapter 7

Conclusions

7.1 Conclusions and Contributions

In the numerous experiments we have conducted, we have reached several general conclusions. First, we have found that clean broadcast news or clean data of any kind is tremendously beneficial, even in small amounts. The intelligibility benefits from even just five minutes of high-quality broadcast news or TTS data makes such data worth seeking out. We have also found that audiobook data also bears many similarities to TTS data, and can be used to create voices with good naturalness and intelligibility.

We have shown that data collected for ASR can be used with a fair amount of cleanup for noise and disfluencies. This type of data is best used by identifying subsets of the best speakers, with ASR word error rate per speaker being a good measure of usability of the data. Again, this type of data is best augmented with even a small amount of high-quality data – the combination of quantity and quality gives intelligibility improvements over using each type of data by itself. While we did not have a great deal of success building voices out of the BABEL data, these corpora were nevertheless useful for their pronunciation lexica, which provided the basis for building frontends for these new languages.

We have explored different methods for training voices that can improve naturalness and intelligibility. Significantly more intelligible voices can be produced from noisy, mixed ASR data using the training data selection approach, especially when selection is done at the speaker level rather than the utterance level. Data selection can also improve naturalness of voices trained on professional broadcast news speech, as can the approach of labeling acoustic and prosodic features at the frontend.

By comparing TTS data to other types of speech data, we have identified the ways in which TTS data distinguishes itself: it tends to have a lower f0 range, a lower standard deviation of energy, and a lower level of articulation than other types of readily-available speech data such as radio broadcast news, audiobooks, and ASR data. We then determined which of these characteristics are informative in producing intelligible and natural-sounding TTS voices. We have found that selecting utterances with a slow speaking rate is in fact detrimental to intelligibility, especially when using conversational ASR data. We have also found that a consistent mean energy level improves intelligibility, which also corresponds with a low overall standard deviation in energy. Identifying the most intelligible speakers in a large, multi-speaker corpus also results in a more intelligible TTS voice. In terms of naturalness, training voices based on low levels of articulation produced more natural-sounding voices for English; however for voices trained on Turkish broadcast news data, the higher setting for articulation was preferred. Low mean f0 was also a consistently informative feature in our naturalness experiments, matching our original observations that TTS speakers tend to speak in a lower f0 range.

Throughout these experiments, we identified which parts of the process present the greatest challenges. First of all, building a linguistic frontend for a new language is one of the more labor-intensive parts of the process, and is necessarily very language-dependent. While for some languages, a very basic frontend consisting mainly of just the pronunciation lexicon may be sufficient, other languages require more complex linguistic processing. For instance, the Amharic language has contrastive gemination and epenthesis which are not written in the orthography, and this has proven to be an entirely open research problem of its own. Even languages which do not require complex linguistic processing still require a phoneset specification, as well as letter-to-sound and syllabification rules for out-of-vocabulary words, which can still take a considerable amount of manual effort to create and check. There has been some success in using the orthography in place of a defined phoneset (Black and Llitjós, 2002; Aylett *et al.*, 2009), but this approach is better suited to languages (such as Spanish) which have a more straightforward mapping between orthography and pronunciation, than those (such as English) which do not.

Another major challenge in our pipeline was evaluation. While Amazon Mechanical Turk was very useful for evaluating English voices, we ran into the limitations of that platform when we expanded to building voices for low-resource languages. While automatic evaluations can provide some guidance with respect to which voices may be viable or not, there is no real substitute for human judgment when it comes to listening for naturalness, or for preference in general. Amazon has implemented more restrictions on new users signing up for Mechanical Turk, such as a months-long waiting times, and a lengthy and complicated identity verification process for those who lack a US social security number or bank account. So, even though we may know speakers of our languages of interest who want to do our listening tests, they have been unable to sign up as workers. In light of this fact, we have begun building our own evaluation website where we can invite people to do our listening tests, and hopefully this will streamline the evaluation process for future experiments.

7.2 Recommendations for Building Voices with Found Data

The large number of experiments we have conducted on building voices with found data has enabled us to reach a variety of conclusions about what worked and did not work. We can formulate a number of recommendations for anyone who might want to build voices on found data, especially in a low-resource language. These recommendations are intended for anyone who might not have the resources or corporate backing to build a production-quality voice, but may nevertheless have access to found data from the web or other research speech corpora. These recommendations may also be useful to anyone who speaks a language that does not already have a TTS system. The availability of open-source tools has made the creation of custom TTS voices more accessible to everyone. It is also preferable that anyone building a voice have some first-hand knowledge of the language, or at least someone whom they can consult about the language, for informal evaluation purposes, especially given that much of the preparation of the frontends did require knowledge of the language.

We found that the intelligibility of voices trained on ASR data only was very low. Even though improvements can be obtained with speaker-level data selection, these voices are still quite difficult to understand. The improvements obtained by adding even a very small amount of higher-quality data make it worthwhile to try to find such data, or even to record it, even if it is only five minutes. Conversely, if only a small amount of high-quality data is available, it may be worthwhile to obtain a larger amount of lower-quality data as well, since this can also give an improvement.

For a TTS developer who is starting with no data at all, it is recommended to try to find radio broadcast news, since this is available in many different languages, and produces relatively intelligible voices. If a medium-sized amount of high-quality data such as broadcast news is available, e.g. a few hours, then the approach of labeling single features at the frontend can result in significant improvement to naturalness, especially when considering mean f0 and articulation level.

We have documented much of the process of preparing new sources of data and recipes for training voices on our website at `http://www.cs.columbia.edu/~ecooper/tts`. We hope that this will be a useful resource for others who wish to explore training voices on found data.

7.3 Directions for Future Work

The aim of this work was to develop methods for training voices on found data that are extensible to low-resource languages. While we have conducted some preliminary experiments on Turkish and Amharic, we have just begun to scratch the surface of exploring the extent to which our results generalize. Our most promising results (speaker selection from a large ASR corpus; quality adaptation using mixed high- and low-quality data; frontend labeling of acoustic and prosodic features) should be repeated on a wide variety of languages to determine which are truly language-independent. Furthermore, all of our experiments were conducted on female data only, so further experimentation is needed to determine which results generalize across gender as well.

While we explored a standard set of acoustic and prosodic features at the utterance and speaker level, there are a number of other features that could potentially be explored, as well as different granularities. For example, low levels of spectral tilt (the proportion of energy at high vs. low frequency bands) is often associated with Lombard speech (speech produced in a noisy environment for the purpose of becoming more intelligible) (Van Summers *et al.*, 1988). Thus, spectral tilt may also be a useful alternative measure of hyper-articulation, and possibly an informative feature for training data subset selection. Furthermore, although we increased our level of granularity from selecting at the utterance level to selecting at the speaker level, we did not try *decreasing* the granularity, e.g. by conducting data selection at the word, phoneme, or acoustic frame level. Indeed, it may be wasteful to discard entire utterances when perhaps only part of an utterance is problematic. There is precedent for this in (Henter *et al.*, 2016), where naturalness was improved by identifying and excluding phonemes that were outliers with respect to duration.

Most of our experiments considered only one feature at a time, and, when we considered multiple features, it was a combination based on best results from a previous round of experiments. An interesting direction for future work would be to use more machine learning or automatic clustering-based approaches for selecting training data subsets, based on multiple features at a time. (Watts *et al.*, 2013) used an active learning based approach to discover sentences similar to a small set of audiobook utterances hand-labeled as ‘neutral’ by a human listener; perhaps we could explore a similar approach where we are measuring similarity to a small set of high-quality TTS data, even in a different language.

Our positive results for combining low- and high-quality data and then performing quality adaptation point to many possibilities for future work in this area. First of all, we could combine quality adaptation with speaker-level data selection, to select our low-quality data in a way that it better matches the speaker characteristics of our small amount of high-quality data. Second, it would also be very interesting to explore whether cross-lingual quality adaptation is possible, i.e. when the low- and high-quality data sets are in different languages. Finally, it is reasonable to ask whether it was the *adaptation* that caused the improvements, or just the act of combining the two different types of data; this is a question for future work as well.

There are some signal processing based approaches that may be worth exploring in future work. Noise removal techniques such as Wiener filtering may succeed in cleaning up the background and line noise in ASR corpora recorded over the telephone, and recent advances in using neural networks for audio super-resolution (Kuleshov *et al.*, 2017) may be applicable to this sort of low-bandwidth data as well.

The end-to-end approach to speech synthesis aims to replace the modular elements of a TTS system with one single, usually neural network based, model that converts raw text directly into raw audio. Systems that approach end-to-end synthesis have become state of the art in recent years (van den Oord *et al.*, 2016; Sercan *et al.*, 2017; Sotelo *et al.*, 2017; Wang *et al.*, 2017), and thus it is important to consider whether our methods are usable in an end-to-end setting. While these systems typically require large amounts of production quality data, the use of “global style tokens” (GST) (Wang *et al.*, 2018) has produced some very interesting results in terms of identifying latent variation in found data. GSTs are unsupervised embeddings that are learned from the audio, which then can be applied to text to synthesize by conditioning the text encoder on one or more tokens. When applied to a large corpus of audiobook data from a single speaker, GSTs were found to encode various acoustic and prosodic attributes, such as pitch, intensity, speaking rate, and emotion. Furthermore, when GSTs were applied to clean speech data with artificially added noise, they encoded different noise conditions, including the condition of no noise. Finally, when applied to TED talk speech from hundreds of different speakers found on the web, GSTs corresponded to different speakers. These experiments show the promise of GSTs for a variety of applications such as prosody and style modeling, noise removal, and speaker diarization, and it would be very interesting to study how this approach could be used to create high-quality voices from other types of found data as well.

There may also be other interesting applications for our approaches. In particular, our frontend-labeling approach allows for control of acoustic and prosodic features which result in audible differences in voices, which may be useful for creating voices with different characteristics. (Levitan *et al.*, 2018) found that many of the same features that we have been exploring for TTS also correlate with whether a listener finds a speaker to be trustworthy or untrustworthy. Thus, we have begun to apply these findings along with our frontend-labeling approach to create voices that have the characteristics of trustworthy and untrustworthy speech. While these voices do sound different from each other, it remains to be seen in future evaluations whether listeners trust or distrust these synthetic voices.

Given the improved naturalness of voices trained with our frontend-labeling approach, it would be interesting to extend these experiments by using actual numeric feature values as the new frontend features rather than discretized settings of high, medium, and low. While HMM-based synthesis does not permit continuous numeric features at the frontend, neural network based synthesis in fact requires this type of input. In terms of other possible frontend features, it would be interesting to see whether the inclusion of prosodic annotations, including those that may have been automatically obtained such as in AuToBI (Rosenberg, 2010), help to improve naturalness of voices trained on found data as well.

Part I

Bibliography

Bibliography

Gopala Krishna Anumanchipalli and Alan W Black. Adaptation techniques for speech synthesis in under-resourced languages. *Spoken Languages Technologies for Under-Resourced Languages*, 2010.

Matthew Aylett, Simon King, and Junichi Yamagishi. Speech synthesis without a phone inventory. *INTERSPEECH*, 2009.

Pallavi Baljekar and Alan W. Black. Utterance selection techniques for TTS systems using found speech. *9th ISCA Speech Synthesis Workshop*, 2016.

Jared Bernstein, Kelsey Taussig, and Jack Godfrey. MACROPHONE: An American English telephone speech corpus for the POLYPHONE project. *ICASSP*, 1994.

Fadi Biadsy, Julia Hirschberg, Andrew Rosenberg, and Wisam Dakka. Comparing American and Palestinian perceptions of charisma using acoustic-prosodic and lexical analysis. *INTERSPEECH*, 2007.

Fadi Biadsy, Andrew Rosenberg, Rolf Carlson, Julia Hirschberg, and Eva Strangert. A cross-cultural comparison of American, Palestinian, and Swedish perception of charismatic speech. *Speech Prosody*, 2008.

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

Alan W. Black and Kevin A. Lenzo. Building synthetic voices - for FestVox 2.0 edition. http://www.festvox.org/bsv, 2003.

Alan W Black and Ariadna Font Llitjós. Unit selection without a phoneme set. In *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*, pages 207–210. IEEE, 2002.

Alan W Black and Keiichi Tokuda. The Blizzard Challenge 2005: Evaluating corpus-based speech synthesis on common databases. *INTERSPEECH*, 2005.

Alan Black, Paul Taylor, and Richard Caley. The Festival speech synthesis system, system documentation, edition 2.4, for Festival version 2.4.0. http://www.festvox.org/docs/manual-2.4.0/festival_toc.html, 2014.

Alan W. Black. The CMU Wilderness multilingual speech dataset. https://github.com/festvox/datasets-CMU_Wilderness, 2018.

Paul Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345, 2001.

Bajibabu Bollepalli, Manu Airaksinen, and Paavo Alku. Lombard speech synthesis using long short-term memory recurrent neural networks. *ICASSP*, 2017.

Norbert Braunschweiler and Sabine Buchholz. Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality. *INTERSPEECH*, 2011.

Sabine Buchholz, Javier Latorre, and Kayoko Yanagisawa. Crowdsourced assessment of speech synthesis. In Maxine Eskénazi, Gina-Anne Levow, Helen Meng, Gabriel Parent, and David Suendermann, editors, *Crowdsourcing for Speech Processing: Applications to Data, Collection, Transcription and Assessment*, chapter 7, pages 173–214. John Wiley & Sons, Ltd, Chichester, 2013.

Alexandra Canavan, David Graff, and George Zipperlen. CALLHOME American English speech corpus LDC97S42. *DVD*, 1997.

Aimilios Chalamandaris, Pirros Tsiakoulis, Sotiris Karabetsos, and Spyros Raptis. Using audio books for training a text-to-speech system. *LREC*, 2014.

Erica Cooper and Julia Hirschberg. Adaptation and frontend features to improve naturalness in found-data synthesis. *Speech Prosody*, 2018.

Erica Cooper, Alison Chang, Yocheved Levitan, and Julia Hirschberg. Data selection and adaptation for naturalness in HMM-based speech synthesis. *INTERSPEECH*, 2016.

Erica Cooper, Yocheved Levitan, and Julia Hirschberg. Data selection for naturalness in HMM-based speech synthesis. *Speech Prosody*, 2016.

Erica Cooper, Xinyue Wang, Alison Chang, Yocheved Levitan, and Julia Hirschberg. Utterance selection for optimizing intelligibility of TTS vocies trained on ASR data. *INTERSPEECH*, 2017.

Erica Cooper, Emily Li, and Julia Hirschberg. Characteristics of text-to-speech and other corpora. *Speech Prosody*, 2018.

Rasmus Dall, Christophe Veaux, Junichi Yamagishi, and Simon King. Analysis of speaker clustering strategies for HMM-based speech synthesis. *INTERSPEECH*, 2012.

Nivja H. de Jong and Ton Wempe. Praat script to detect syllable nuclei and measure speech rate automatically. *Behavior Research Methods*, 41(2):385–390, 2009.

Nivja H. de Jong, Rachel Groenhout, Rob Schoonen, and Jan H. Hulstijn. Second language fluency: Speaking style or proficiency? correcting measures of second language fluency for first language behavior. *Applied Psycholinguistics*, 36(2):223–243, 2015.

Jelske Dijkstra, Louis C.W. Pols, and R.J.J.H. van Son. Frisian TTS, an example of bootstrapping TTS for minority languages. *5th ISCA Speech Synthesis Workshop*, 2004.

Moses Ekpenyong, Eno-Abasi Urua, Oliver Watts, Simon King, and Junichi Yamagishi. Statistical parametric speech synthesis for Ibibio. *Speech Communication*, 56:243–251, 2014.

A. Gallardo-Antolín, J.M. Montero, and S. King. A comparison of open-source segmentation architectures for dealing with imperfect data from the media in speech synthesis. *INTERSPEECH*, 2014.

Kallirroi Georgila, Alan W. Black, Kenji Sagae, and David Traum. Practical evaluation of human and synthesized speech for virtual human dialogue systems. *LREC*, 2012.

Kyle Gorman, Jonathan Howell, and Michael Wagner. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193, 2011.

Avashna Govender and Febe de Wet. Objective measures to improve the selection of training speakers in HMM-based child speech synthesis. *PRASA-RobMech*, 2016.

Alexander Gutkin, Linne Ha, Martin Jansche, Knot Pipatsrisawat, and Richard Sproat. TTS for low resource languages: A Bangla synthesizer. *Language Resources and Evaluation Conference*, 2016.

Mary Harper. BABEL: IARPA solicitation IARPA-BAA-11-02. 2011.

Gustav Eje Henter, Srikanth Ronanki, Oliver Watts, Mirjam Wester, Zhizheng Wu, and Simon King. Robust TTS duration modeling using DNNs. *ICASSP*, 2016.

Gustav Eje Henter, Jaime Lorenzo-Trueba, Xin Wang, and Junichi Yamagishi. Principles for learning controllable TTS from annotated and latent variation. *INTERSPEECH*, 2017.

Julia Hirschberg and Mary Beckman. The ToBI annotation conventions. 1994.

Shiyin Kang, Xiaojun Qian, and Helen Meng. Multi-distribution deep belief network for speech synthesis. *ICASSP*, 2013.

Reima Karhila, Ulpu Remes, and Mikko Kurimo. HMM-based speech synthesis adaptation using noisy data: Analysis and evaluation methods. *ICASSP*, 2013.

John Kominek and Alan W Black. CMU Arctic databases for speech synthesis. Technical report, 2003.

John Kominek, Tanja Schultz, and Alan W. Black. Voice building from insufficient data – classroom experiences with web-based language development tools. *6th ISCA Speech Synthesis Workshop*, 2007.

Robert F. Kubichek. Mel-cepstral distance measure for objective speech quality assessment. *Proc. IEEE Pacific Rim Conf. Communications, Computers, and Signal Processing*, 1993.

Volodymyr Kuleshov, S. Zayd Enam, and Stefano Ermon. Audio super-resolution using neural nets. *ICLR (Workshop Track)*, 2017.

Kai-Zhan Lee, Erica Cooper, and Julia Hirschberg. A comparison of speaker-based and utterance-based data selection for text-to-speech synthesis. *INTERSPEECH*, 2018.

Sarah Ita Levitan, Angel Maredia, and Julia Hirschberg. Acoustic-prosodic indicators of deception and trust in interview dialogues. *INTERSPEECH*, 2018.

Sebsibe H Mariam, S P Kishore, Alan W Black, Rohit Kumar, and Rajeev Sangal. Unit selection voice for Amharic using Festvox. *Fifth ISCA Workshop on Speech Synthesis*, 2004.

Jindřich Matoušek, Daniel Tihelka, and Jan Romportl. Building of a speech corpus optimised for unit selection TTS synthesis. *LREC*, 2008.

Thomas Merritt, Javier Latorre, and Simon King. Attributing modelling errors in HMM synthesis by stepping gradually from natural to modelled speech. *ICASSP*, 2015.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD: A vocoder-based high quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, 2016.

NIST. Speech quality assurance (SPQA) package version 2.3 and speech file manipulation software (SPHERE) package version 2.5. 2009.

Mari Ostendorf, Patti J. Price, and Stefanie Shattuck-Hufnagel. The Boston University radio news corpus. *Tech. Rep.*, 1995.

E. Pavlick, M. Post, A. Irvine, D. Kachaev, and C. Callison-Burch. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, pages 79–92, 2014.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hanneman, Petr Motlicek, Yanmin Qian, Petr Schwartz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

Kishore Prahallad, Alan W. Black, and Ravishankhar Mosur. Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis. *ICASSP*, 2006.

Kishore Prahallad, E Naresh Kumar, Venkatesh Keri, S Rajendran, and Alan W Black. The IIIT-H Indic speech databases. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

Andrew Rosenberg and Julia Hirschberg. Acoustic/prosodic and lexical correlates of charismatic speech. *Eurospeech*, 2005.

Andrew Rosenberg and Julia Hirschberg. Charisma perception from text and speech. *Speech Communication*, 2008.

Andrew Rosenberg. AuToBI - a tool for automatic ToBI annotation. *INTERSPEECH*, 2010.

M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier. An open-source state-of-the-art toolbox for broadcast news diarization. *INTERSPEECH*, 2013.

Murat Saraçlar. Turkish broadcast news speech and transcripts LDC2012S06. *DVD*, 2012.

Astrid Schmidt-Nielsen. Intelligibility and acceptability testing for speech technology. *No. NRL/FR/5530-92-9379. Naval Research Lab*, 1992.

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.

Arik Sercan, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2017.

Sunayana Sitaram, Gopala Krishna Anumanchipalli, Justin Chiu, Alok Parlikar, and Alan W Black. Text to speech in new languages without a standardized orthography. *8th ISCA Speech Synthesis Workshop*, 2013.

Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. *ICLR*, 2017.

A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. A. J. Clark, J. Yamagishi, and S. King. TUNDRA: a multilingual corpus of found data for TTS research created with light supervision. *INTERSPEECH*, 2013.

Adriana Stan, Florina Dinescu, Cristina Țiple, Șerban Meza, Bogdan Orza, Magdalena Chirilă, and Mircea Giurgiu. The SWARA speech corpus: A large parallel romanian read speech dataset. *International Conference on Speech Technology and Human-Computer Dialogue*, 2017.

Pawel Swietojanski, Jinyu Li, and Steve Renals. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1450–1463, 2016.

Cassia Valentini-Botinhao, Junichi Yamagishi, and Simon King. Evaluation of objective measures for intelligibility prediction of HMM-based synthetic speech in noise. *ICASSP*, 2011.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

Walter Van Summers, David B. Pisoni, Robert H. Bernacki, Robert I. Pedlow, and Michael A. Stokes. Effects of noise on speech production: Acoustic and perceptual analyses. *The Journal of the Acoustical Society of America*, 84(3):917–928, 1988.

D. Voiers. Diagnostic acceptability measure for speech communication systems. *Proc. Int. Conf. on Acoust., Speech, Signal Processing*, 1977.

X. Wang, S. Takaki, and J. Yamagishi. A comparative study of the performance of HMM, DNN, and RNN based speech synthesis systems trained on very large speaker-dependent corpora. *9th ISCA Speech Synthesis Workshop*, 2016.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*, 2018.

O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King. Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis. *8th ISCA Speech Synthesis Workshop*, 2013.

Oliver Watts, Gustav Eje Henter, Thomas Merritt, Zhizheng Wu, and Simon King. From HMMs to DNNs: Where do the improvements come from? *ICASSP*, 2016.

Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An open source neural network speech synthesis system. *9th ISCA Speech Synthesis Workshop*, 2016.

Junichi Yamagishi, Koji Onishi, Takashi Masuko, and Takao Kobayashi. Modeling of various speaking styles and emotions for HMM-based speech synthesis. *EUROSPEECH*, 2003.

Junichi Yamagishi, Zhenhua Ling, and Simon King. Robustness of HMM-based speech synthesis. *INTERSPEECH*, 2008.

Junichi Yamagishi, Bela Usabaev, Simon King, Oliver Watts, John Dines, Jilei Tian, Yong Guan, Rile Hu, Keiichiro Oura, Yi-Jian Wu, Keiichi Tokuda, Reima Karhila, and Mikko Kurimo. Thousands of voices for HMM-based speech synthesis - analysis and application of TTS systems built on various ASR corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5), 2010.

Junichi Yamagishi. *Average-voice-based Speech Synthesis*. PhD thesis, Tokyo Institute of Technology, 2006.

Hongwu Yang, Keiichiro Oura, Haiyan Wang, Zhenye Gan, and Keiichi Tokuda. Using speaker adaptive training to realize Mandarin-Tibetan cross-lingual speech synthesis. *Multimedia Tools and Applications*, 74(22):9927–9942, 2015.

S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3)*. Cambridge University, 1995.

Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W. Black, and Keiichi Tokuda. The HMM-based speech synthesis system (HTS) version 2.0. *6th ISCA Workshop on Speech Synthesis*, 2007.

Heiga Zen, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. *ICASSP*, 2013.

Part II

Appendices

Appendix A

HTS Full-Context Label Format

The following page contains a description of the full-context phonetic label format for HTS and Merlin, with a description of each of the contextual features, reproduced from the HTS (Zen *et al.*, 2007) documentation.

An example of context-dependent label format for HMM-based speech synthesis in English

HTS Working Group

December 25, 2015

$p_1\text{\^{}}p_2\text{-}p_3\text{+}p_4\text{=}p_5\texttt{@}p_6\_p_7$

$\texttt{/A:}a_1\_a_2\_a_3$ $\texttt{/B:}b_1\text{-}b_2\text{-}b_3\texttt{@}b_4\text{-}b_5\texttt{\&}b_6\text{-}b_7\texttt{\#}b_8\text{-}b_9\texttt{\$}b_{10}\text{-}b_{11}\texttt{!}b_{12}\text{-}b_{13}\texttt{;}b_{14}\text{-}b_{15}\texttt{|}b_{16}$ $\texttt{/C:}c_1\text{+}c_2\text{+}c_3$

$\texttt{/D:}d_1\_d_2$ $\texttt{/E:}e_1\text{+}e_2\texttt{@}e_3\text{+}e_4\texttt{\&}e_5\text{+}e_6\texttt{\#}e_7\text{+}e_8$ $\texttt{/F:}f_1\_f_2$

$\texttt{/G:}g_1\_g_2$ $\texttt{/H:}h_1\text{=}h_2\text{\^{}}h_3\text{=}h_4\texttt{|}h_5$ $\texttt{/I:}i_1\text{=}i_2$

$\texttt{/J:}j_1\text{+}j_2\text{-}j_3$

p_1	the phoneme identity before the previous phoneme
p_2	the previous phoneme identity
p_3	the current phoneme identity
p_4	the next phoneme identity
p_5	the phoneme after the next phoneme identity
p_6	position of the current phoneme identity in the current syllable (forward)
p_7	position of the current phoneme identity in the current syllable (backward)
a_1	whether the previous syllable stressed or not (0: not stressed, 1: stressed)
a_2	whether the previous syllable accented or not (0: not accented, 1: accented)
a_3	the number of phonemes in the previous syllable
b_1	whether the current syllable stressed or not (0: not stressed, 1: stressed)
b_2	whether the current syllable accented or not (0: not accented, 1: accented)
b_3	the number of phonemes in the current syllable
b_4	position of the current syllable in the current word (forward)
b_5	position of the current syllable in the current word (backward)
b_6	position of the current syllable in the current phrase (forward)
b_7	position of the current syllable in the current phrase (backward)
b_8	the number of stressed syllables before the current syllable in the current phrase
b_9	the number of stressed syllables after the current syllable in the current phrase
b_{10}	the number of accented syllables before the current syllable in the current phrase
b_{11}	the number of accented syllables after the current syllable in the current phrase
b_{12}	the distance per syllable from the previous stressed syllable to the current syllable
b_{13}	the distance per syllable from the current syllable to the next stressed syllable
b_{14}	the distance per syllable from the previous accented syllable to the current syllable
b_{15}	the distance per syllable from the current syllable to the next accented syllable
b_{16}	name of the vowel of the current syllable
c_1	whether the next syllable stressed or not (0: not stressed, 1: stressed)
c_2	whether the next syllable accented or not (0: not accented, 1: accented)
c_3	the number of phonemes in the next syllable
d_1	gpos (guess part-of-speech) of the previous word
d_2	the number of syllables in the previous word
e_1	gpos (guess part-of-speech) of the current word
e_2	the number of syllables in the current word
e_3	position of the current word in the current phrase (forward)
e_4	position of the current word in the current phrase (backward)
e_5	the number of content words before the current word in the current phrase
e_6	the number of content words after the current word in the current phrase
e_7	the distance per word from the previous content word to the current word
e_8	the distance per word from the current word to the next content word
f_1	gpos (guess part-of-speech) of the next word
f_2	the number of syllables in the next word
g_1	the number of syllables in the previous phrase
g_2	the number of words in the previous phrase
h_1	the number of syllables in the current phrase
h_2	the number of words in the current phrase
h_3	position of the current phrase in this utterance (forward)
h_4	position of the current phrase in this utterance (backward)
h_5	TOBI endtone of the current phrase
i_1	the number of syllables in the next phrase
i_2	the number of words in the next phrase
j_1	the number of syllables in this utterance
j_2	the number of words in this utterance
j_3	the number of phrases in this utterance

Appendix B

Full Results for Macrophone Speaker-vs. Utterance-Selected Voice Experiments

These are the full preliminary results for word error rate using the Watson Speech Recognition API.

B.1 Speaker-Selected Voices

Voices are named are as follows:

`macrophone-[selectionfeature]-[selectionlevel]-[subsetsize]`

Average word error rates (WER) from transcriptions given by the Watson Speech Recognition API are presented as values between 0 and 1, as well as average confidence values.

Voice	WER	Confidence
macrophone-baseline	0.818	0.488
macrophone-eng_max-avg-10hr	0.662	0.549
macrophone-eng_max-avg-2hr	0.844	0.554
macrophone-eng_max-avg-4hr	0.779	0.503
macrophone-eng_max-max-10hr	1.013	0.410
macrophone-eng_max-max-2hr	0.987	0.392
macrophone-eng_max-max-4hr	0.974	0.426
macrophone-eng_max-med-10hr	0.636	0.535
macrophone-eng_max-med-2hr	0.883	0.442
macrophone-eng_max-med-4hr	0.805	0.553
macrophone-eng_max-min-10hr	0.662	0.583
macrophone-eng_max-min-2hr	0.857	0.501
macrophone-eng_max-min-4hr	0.740	0.561
macrophone-eng_mean-avg-10hr	0.740	0.557
macrophone-eng_mean-avg-2hr	0.766	0.497
macrophone-eng_mean-avg-4hr	0.727	0.510
macrophone-eng_mean-max-10hr	0.831	0.508
macrophone-eng_mean-max-2hr	1.026	0.453
macrophone-eng_mean-max-4hr	1.143	0.360
macrophone-eng_mean-med-10hr	0.714	0.539
macrophone-eng_mean-med-2hr	0.805	0.441
macrophone-eng_mean-med-4hr	0.831	0.497
macrophone-eng_mean-min-10hr	0.610	0.512
macrophone-eng_mean-min-2hr	0.909	0.508
macrophone-eng_mean-min-4hr	0.779	0.511
macrophone-eng_min-avg-10hr	0.727	0.556
macrophone-eng_min-avg-2hr	0.896	0.483
macrophone-eng_min-avg-4hr	0.870	0.435
macrophone-eng_min-max-10hr	0.961	0.415
macrophone-eng_min-max-2hr	1.000	0.486
macrophone-eng_min-max-4hr	1.000	0.370
macrophone-eng_min-med-10hr	0.714	0.530
macrophone-eng_min-med-2hr	0.818	0.486
macrophone-eng_min-med-4hr	0.740	0.520
macrophone-eng_min-min-10hr	0.623	0.567
macrophone-eng_min-min-2hr	0.870	0.460

macrophone-eng_min-min-4hr	0.779	0.467
macrophone-eng_stdv-avg-10hr	0.779	0.496
macrophone-eng_stdv-avg-2hr	0.870	0.515
macrophone-eng_stdv-avg-4hr	0.766	0.478
macrophone-eng_stdv-max-10hr	0.662	0.503
macrophone-eng_stdv-max-2hr	0.857	0.475
macrophone-eng_stdv-max-4hr	0.818	0.466
macrophone-eng_stdv-med-10hr	0.766	0.546
macrophone-eng_stdv-med-2hr	0.857	0.496
macrophone-eng_stdv-med-4hr	0.831	0.494
macrophone-eng_stdv-min-10hr	0.831	0.477
macrophone-eng_stdv-min-2hr	0.974	0.269
macrophone-eng_stdv-min-4hr	1.000	0.338
macrophone-f0_mas-avg-10hr	0.610	0.551
macrophone-f0_mas-avg-2hr	0.883	0.405
macrophone-f0_mas-avg-4hr	0.779	0.540
macrophone-f0_mas-max-10hr	0.675	0.557
macrophone-f0_mas-max-2hr	0.792	0.432
macrophone-f0_mas-max-4hr	0.766	0.493
macrophone-f0_mas-med-10hr	0.870	0.426
macrophone-f0_mas-med-2hr	0.870	0.400
macrophone-f0_mas-med-4hr	0.805	0.502
macrophone-f0_mas-min-10hr	0.870	0.569
macrophone-f0_mas-min-2hr	1.000	0.435
macrophone-f0_mas-min-4hr	0.987	0.330
macrophone-f0_max-avg-10hr	0.740	0.500
macrophone-f0_max-avg-2hr	0.870	0.419
macrophone-f0_max-avg-4hr	0.883	0.509
macrophone-f0_max-max-10hr	0.766	0.516
macrophone-f0_max-max-2hr	1.000	0.394
macrophone-f0_max-max-4hr	0.870	0.446
macrophone-f0_max-med-10hr	0.818	0.442
macrophone-f0_max-med-2hr	1.000	0.424
macrophone-f0_max-med-4hr	0.909	0.412
macrophone-f0_max-min-10hr	0.779	0.527
macrophone-f0_max-min-2hr	0.883	0.447
macrophone-f0_max-min-4hr	0.779	0.452

macrophone-f0_mean-avg-10hr	0.766	0.497
macrophone-f0_mean-avg-2hr	0.857	0.459
macrophone-f0_mean-avg-4hr	0.818	0.491
macrophone-f0_mean-max-10hr	0.805	0.495
macrophone-f0_mean-max-2hr	1.000	0.379
macrophone-f0_mean-max-4hr	0.935	0.408
macrophone-f0_mean-med-10hr	0.727	0.518
macrophone-f0_mean-med-2hr	0.909	0.444
macrophone-f0_mean-med-4hr	0.792	0.494
macrophone-f0_mean-min-10hr	0.675	0.503
macrophone-f0_mean-min-2hr	0.831	0.380
macrophone-f0_mean-min-4hr	0.831	0.451
macrophone-f0_median-avg-10hr	0.714	0.579
macrophone-f0_median-avg-2hr	0.961	0.447
macrophone-f0_median-avg-4hr	0.857	0.511
macrophone-f0_median-max-10hr	0.714	0.463
macrophone-f0_median-max-2hr	0.922	0.409
macrophone-f0_median-max-4hr	1.000	0.419
macrophone-f0_median-med-10hr	0.675	0.529
macrophone-f0_median-med-2hr	0.844	0.448
macrophone-f0_median-med-4hr	0.753	0.547
macrophone-f0_median-min-10hr	0.831	0.553
macrophone-f0_median-min-2hr	0.948	0.389
macrophone-f0_median-min-4hr	0.870	0.441
macrophone-f0_min-avg-10hr	0.766	0.538
macrophone-f0_min-avg-2hr	0.909	0.489
macrophone-f0_min-avg-4hr	0.883	0.466
macrophone-f0_min-max-10hr	0.714	0.530
macrophone-f0_min-max-2hr	0.961	0.425
macrophone-f0_min-max-4hr	0.792	0.508
macrophone-f0_min-med-10hr	0.727	0.484
macrophone-f0_min-med-2hr	0.948	0.479
macrophone-f0_min-med-4hr	0.844	0.475
macrophone-f0_min-min-10hr	0.792	0.509
macrophone-f0_min-min-2hr	0.896	0.456
macrophone-f0_min-min-4hr	0.792	0.521
macrophone-f0_stdv-avg-10hr	0.714	0.516

macrophone-f0_stdv-avg-2hr	0.961	0.338
macrophone-f0_stdv-avg-4hr	0.779	0.513
macrophone-f0_stdv-max-10hr	0.779	0.541
macrophone-f0_stdv-max-2hr	0.948	0.354
macrophone-f0_stdv-max-4hr	0.961	0.461
macrophone-f0_stdv-med-10hr	0.688	0.532
macrophone-f0_stdv-med-2hr	0.909	0.435
macrophone-f0_stdv-med-4hr	0.844	0.520
macrophone-f0_stdv-min-10hr	0.688	0.508
macrophone-f0_stdv-min-2hr	1.026	0.490
macrophone-f0_stdv-min-4hr	0.818	0.452
macrophone-speech_rate-avg-10hr	0.792	0.516
macrophone-speech_rate-avg-2hr	0.883	0.520
macrophone-speech_rate-avg-4hr	0.831	0.543
macrophone-speech_rate-max-10hr	0.714	0.545
macrophone-speech_rate-max-2hr	0.909	0.543
macrophone-speech_rate-max-4hr	0.753	0.507
macrophone-speech_rate-med-10hr	0.688	0.572
macrophone-speech_rate-med-2hr	0.870	0.475
macrophone-speech_rate-med-4hr	0.779	0.501
macrophone-speech_rate-min-10hr	0.909	0.471
macrophone-speech_rate-min-2hr	1.026	0.340
macrophone-speech_rate-min-4hr	1.143	0.403
macrophone-vcd2tot_frames-avg-10hr	0.727	0.503
macrophone-vcd2tot_frames-avg-2hr	0.792	0.444
macrophone-vcd2tot_frames-avg-4hr	0.766	0.513
macrophone-vcd2tot_frames-max-10hr	0.597	0.511
macrophone-vcd2tot_frames-max-2hr	0.870	0.482
macrophone-vcd2tot_frames-max-4hr	0.870	0.510
macrophone-vcd2tot_frames-med-10hr	0.662	0.507
macrophone-vcd2tot_frames-med-2hr	0.805	0.508
macrophone-vcd2tot_frames-med-4hr	0.818	0.517
macrophone-vcd2tot_frames-min-10hr	0.831	0.548
macrophone-vcd2tot_frames-min-2hr	0.987	0.437
macrophone-vcd2tot_frames-min-4hr	1.000	0.424
macrophone-WER-avg-10hr	0.766	0.468
macrophone-WER-avg-2hr	0.961	0.365

macrophone-WER-avg-4hr	0.805	0.531
macrophone-WER-max-10hr	0.844	0.483
macrophone-WER-max-2hr	0.922	0.401
macrophone-WER-max-4hr	0.922	0.378
macrophone-WER-med-10hr	0.701	0.532
macrophone-WER-med-2hr	0.857	0.510
macrophone-WER-med-4hr	0.779	0.488
macrophone-WER-min-10hr	0.584	0.537
macrophone-WER-min-2hr	0.870	0.463
macrophone-WER-min-4hr	0.792	0.482

B.2 Utterance-Selected Voices

Voices are named are as follows:

```
macrophone\_utt-[selectionfeature]-[selectionlevel]-10hr
```

All voices are trained on 10 hours of data, as it was seen from the utterance-selected voices that training on 10 hours of data is consistently better.

Voice	**WER**	**Confidence**
macrophone-baseline	0.818	0.488
macrophone_utt-eng_max-avg-10hr	0.779	0.559
macrophone_utt-eng_max-max-10hr	1.013	0.539
macrophone_utt-eng_max-med-10hr	0.779	0.510
macrophone_utt-eng_max-min-10hr	0.753	0.496
macrophone_utt-eng_mean-avg-10hr	0.805	0.496
macrophone_utt-eng_mean-max-10hr	0.675	0.424
macrophone_utt-eng_mean-med-10hr	0.714	0.494
macrophone_utt-eng_mean-min-10hr	1.000	0.461
macrophone_utt-eng_min-avg-10hr	0.818	0.465
macrophone_utt-eng_min-max-10hr	1.039	0.391
macrophone_utt-eng_min-med-10hr	0.805	0.453
macrophone_utt-eng_min-min-10hr	0.805	0.368
macrophone_utt-eng_stdv-avg-10hr	0.805	0.487
macrophone_utt-eng_stdv-max-10hr	0.883	0.497
macrophone_utt-eng_stdv-med-10hr	0.844	0.428
macrophone_utt-eng_stdv-min-10hr	0.987	0.550
macrophone_utt-f0_mas-avg-10hr	0.792	0.494
macrophone_utt-f0_mas-max-10hr	0.974	0.419
macrophone_utt-f0_mas-med-10hr	0.805	0.527
macrophone_utt-f0_mas-min-10hr	0.805	0.556
macrophone_utt-f0_max-avg-10hr	0.844	0.525
macrophone_utt-f0_max-max-10hr	0.740	0.513
macrophone_utt-f0_max-med-10hr	0.870	0.452
macrophone_utt-f0_max-min-10hr	0.844	0.483
macrophone_utt-f0_mean-avg-10hr	0.792	0.509
macrophone_utt-f0_mean-max-10hr	0.857	0.483
macrophone_utt-f0_mean-med-10hr	0.688	0.525
macrophone_utt-f0_mean-min-10hr	0.857	0.462
macrophone_utt-f0_median-avg-10hr	0.766	0.513
macrophone_utt-f0_median-max-10hr	0.948	0.419
macrophone_utt-f0_median-med-10hr	0.805	0.463
macrophone_utt-f0_median-min-10hr	0.831	0.506
macrophone_utt-f0_min-avg-10hr	0.779	0.506
macrophone_utt-f0_min-max-10hr	0.896	0.493
macrophone_utt-f0_min-med-10hr	0.792	0.489

macrophone_utt-f0_min-min-10hr	0.831	0.559
macrophone_utt-f0_stdv-avg-10hr	0.870	0.509
macrophone_utt-f0_stdv-max-10hr	0.844	0.527
macrophone_utt-f0_stdv-med-10hr	0.753	0.422
macrophone_utt-f0_stdv-min-10hr	0.935	0.411
macrophone_utt-speech_rate-avg-10hr	1.104	0.432
macrophone_utt-speech_rate-max-10hr	1.000	0.911
macrophone_utt-speech_rate-med-10hr	0.987	0.480
macrophone_utt-speech_rate-min-10hr	0.740	0.560
macrophone_utt-vcd2tot_frames-avg-10hr	0.753	0.430
macrophone_utt-vcd2tot_frames-max-10hr	0.974	0.433
macrophone_utt-vcd2tot_frames-med-10hr	0.805	0.451
macrophone_utt-vcd2tot_frames-min-10hr	0.948	0.470
macrophone_utt-WER-avg-10hr	0.805	0.453
macrophone_utt-WER-max-10hr	0.870	0.472
macrophone_utt-WER-med-10hr	0.805	0.495
macrophone_utt-WER-min-10hr	1.000	0.428

Appendix C

Custom Question Set for Amharic

```
QS "C-Voiced_lateral"              {*-lw+*,*-l+*}
QS "C-Unvoiced_Plosive"            {*-kw+*,*-tw+*,*-pw+*,*-k+*,*-p+*,*-t+*}
QS "C-Unvoiced_Bilabial"           {*-pw+*,*-p+*}
QS "C-Palatal"                     {*-S+*,*-j+*,*-tSGTw+*,*-Z+*,*-dZw+*,*-Zw+*,*-J+*,*-tSw+*,
                                    *-dZ+*,*-tSGT+*,*-Jw+*,*-tS+*,*-Sw+*}
QS "C-Voiced_Consonant"            {*-j+*,*-Z+*,*-dZw+*,*-bw+*,*-dw+*,*-Zw+*,*-lw+*,*-mw+*,
                                    *-J+*,*-dZ+*,*-gw+*,*-v+*,*-Jw+*,*-vw+*,*-zw+*,*-b+*,
                                    *-rw+*,*-d+*,*-g+*,*-m+*,
                                    *-l+*,*-n+*,*-r+*,*-w+*,*-z+*,*-nw+*}
QS "C-Lateral"                     {*-lw+*,*-l+*}
QS "C-Voiced_Plosive"              {*-bw+*,*-dw+*,*-gw+*,*-b+*,*-d+*,*-g+*}
QS "C-Pulmonic_Voiced_Bilabial"    {*-bw+*,*-mw+*,*-b+*,*-m+*}
QS "C-Pulmonic_Nasal"              {*-mw+*,*-J+*,*-Jw+*,*-m+*,*-n+*,*-nw+*}
QS "C-Unvoiced_Consonant"          {*-S+*,*-fw+*,*-hw+*,*-sw+*,*-kw+*,*-tw+*,*-tSw+*,*-tS+*,
                                    *-pw+*,*-f+*,*-h+*,*-k+*,*-p+*,*-s+*,*-t+*,*-Sw+*}
QS "C-Velar_Plosive"               {*-kGTw+*,*-kw+*,*-kGT+*,*-gw+*,*-g+*,*-k+*}
QS "C-Pulmonic_Bilabial"           {*-bw+*,*-mw+*,*-b+*,*-pw+*,*-m+*,*-p+*}
QS "C-Bilabial_Plosive"            {*-bw+*,*-pGTw+*,*-b+*,*-pw+*,*-p+*}
QS "C-Pulmonic_Coronal"            {*-S+*,*-j+*,*-Z+*,*-dZw+*,*-dw+*,*-Zw+*,*-lw+*,*-sw+*,
                                    *-J+*,*-tw+*,*-tSw+*,*-dZ+*,*-Jw+*,*-zw+*,*-tS+*,*-rw+*,
                                    *-d+*,*-l+*,*-n+*,*-s+*,*-r+*,*-t+*,*-z+*,*-Sw+*,*-nw+*}
QS "C-Front_Vowel"                 {*-a+*,*-e+*,*-i+*}
QS "C-Pulmonic_Approximant"        {*-j+*,*-l+*,*-w+*}
QS "C-Unvoiced_Fricative"          {*-S+*,*-fw+*,*-hw+*,*-sw+*,*-f+*,*-h+*,*-s+*,*-Sw+*}
QS "C-Alveolar_Fricative"          {*-sw+*,*-sGTw+*,*-zw+*,*-sGT+*,*-s+*,*-z+*}
QS "C-Nasal"                       {*-mw+*,*-J+*,*-Jw+*,*-m+*,*-n+*,*-nw+*}
QS "C-Voiced_Coronal"              {*-j+*,*-Z+*,*-dZw+*,*-dw+*,*-Zw+*,*-lw+*,*-J+*,*-dZ+*,
                                    *-Jw+*,*-zw+*,*-rw+*,*-d+*,*-l+*,*-n+*,*-r+*,*-z+*,
                                    *-nw+*}
QS "C-Coronal_Plosive"             {*-dw+*,*-tw+*,*-tGTw+*,*-tGT+*,*-d+*,*-t+*}
QS "C-Pulmonic_Consonant"          {*-S+*,*-j+*,*-fw+*,*-Z+*,*-hw+*,*-dZw+*,*-bw+*,*-dw+*,
                                    *-Zw+*,*-lw+*,*-sw+*,*-mw+*,*-J+*,*-kw+*,*-tw+*,*-tSw+*,
                                    *-dZ+*,*-gw+*,*-v+*,*-Jw+*,*-vw+*,*-zw+*,*-b+*,*-tS+*,
                                    *-pw+*,*-rw+*,*-d+*,*-g+*,*-f+*,*-h+*,*-k+*,
                                    *-m+*,*-l+*,*-n+*,*-p+*,*-s+*,*-r+*,*-t+*,*-w+*,*-z+*,
                                    *-Sw+*,*-nw+*}
QS "C-Coronal"                     {*-S+*,*-j+*,*-tSGTw+*,*-Z+*,*-dZw+*,*-dw+*,*-Zw+*,
                                    *-lw+*,*-sw+*,*-J+*,*-sGTw+*,*-tw+*,*-tSw+*,*-dZ+*,
                                    *-tGTw+*,*-tSGT+*,*-Jw+*,*-zw+*,*-tS+*,*-tGT+*,*-rw+*,
                                    *-d+*,*-sGT+*,*-l+*,*-n+*,*-s+*,*-r+*,*-t+*,*-z+*,
                                    *-Sw+*,*-nw+*}
QS "C-High_Vowel"                  {*-a+*}
QS "C-Vowel"                       {*-at+*,*-a+*,*-e+*,*-i+*,*-o+*,*-u+*,*-one+*}
QS "C-Mid_Vowel"                   {*-at+*,*-e+*,*-o+*}
QS "C-Voiced_Bilabial"             {*-bw+*,*-mw+*,*-b+*,*-m+*}
QS "C-Unrounded"                   {*-at+*,*-a+*,*-e+*,*-i+*,*-one+*}
QS "C-Consonant"                   {*-S+*,*-j+*,*-tSGTw+*,*-fw+*,*-Z+*,*-hw+*,*-dZw+*,
                                    *-bw+*,*-pGTw+*,*-dw+*,*-Zw+*,*-kGTw+*,*-lw+*,*-sw+*,
                                    *-mw+*,*-J+*,*-kw+*,*-sGTw+*,*-tw+*,*-tSw+*,*-dZ+*,
                                    *-tGTw+*,*-kGT+*,*-gw+*,*-v+*,*-tSGT+*,*-Jw+*,*-vw+*,
                                    *-zw+*,*-b+*,*-tS+*,*-pw+*,*-tGT+*,*-rw+*,*-d+*,*-g+*,
                                    *-f+*,*-h+*,*-k+*,*-sGT+*,*-m+*,*-l+*,*-n+*,*-p+*,*-s+*,
                                    *-r+*,*-t+*,*-w+*,*-z+*,*-Sw+*,*-nw+*}
QS "C-Unvoiced_Coronal"            {*-S+*,*-sw+*,*-tw+*,*-tSw+*,*-tS+*,*-s+*,*-t+*,*-Sw+*}
QS "C-Voiced_Fricative"            {*-Z+*,*-Zw+*,*-v+*,*-vw+*,*-zw+*,*-z+*}
QS "C-Velar"                       {*-kGTw+*,*-kw+*,*-kGT+*,*-gw+*,*-g+*,*-k+*}
QS "C-Central_Vowel"               {*-at+*,*-one+*}
QS "C-Approximant"                 {*-j+*,*-l+*,*-w+*}
QS "C-Back_Vowel"                  {*-o+*,*-u+*}
QS "C-silences"                    {*-pau+*,*-wb+*}
QS "C-Postalveolar_Fricative"      {*-S+*,*-Z+*,*-Zw+*,*-Sw+*}
QS "C-Plosive"                     {*-bw+*,*-pGTw+*,*-dw+*,*-kGTw+*,*-kw+*,*-tw+*,*-tGTw+*,
                                    *-kGT+*,*-gw+*,*-b+*,*-pw+*,*-tGT+*,*-d+*,*-g+*,*-k+*,
                                    *-p+*,*-t+*}
QS "C-Palatal_affricate"           {*-tSGTw+*,*-dZw+*,*-tSw+*,*-dZ+*,*-tSGT+*,*-tS+*}
QS "C-Ejective_affricate"          {*-tSGTw+*,*-tSGT+*}
QS "C-Unvoiced_Pulmonic_Consonant" {*-S+*,*-fw+*,*-hw+*,*-sw+*,*-kw+*,*-tw+*,*-tSw+*,*-tS+*,
                                    *-pw+*,*-f+*,*-h+*,*-k+*,*-p+*,*-s+*,*-t+*,*-Sw+*}
QS "C-Fricative"                   {*-S+*,*-fw+*,*-Z+*,*-hw+*,*-Zw+*,*-sw+*,*-sGTw+*,*-v+*,
                                    *-vw+*,*-zw+*,*-f+*,*-h+*,*-sGT+*,*-s+*,*-z+*,*-Sw+*}
QS "C-Rounded"                     {*-o+*,*-u+*}
QS "C-Affricate"                   {*-tSGTw+*,*-dZw+*,*-tSw+*,*-dZ+*,*-tSGT+*,*-tS+*}
QS "C-Voiced_Velar"                {*-gw+*,*-g+*}
QS "C-Voiced_Pulmonic_Consonant"   {*-j+*,*-Z+*,*-dZw+*,*-bw+*,*-dw+*,*-Zw+*,*-lw+*,*-mw+*,
                                    *-J+*,*-dZ+*,*-gw+*,*-v+*,*-Jw+*,*-vw+*,*-zw+*,*-b+*,
                                    *-rw+*,*-d+*,*-g+*,*-m+*,*-l+*,*-n+*,*-r+*,*-w+*,*-z+*,
                                    *-nw+*}
QS "C-Bilabial"                    {*-bw+*,*-pGTw+*,*-mw+*,*-b+*,*-pw+*,*-m+*,*-p+*}
QS "C-Ejective"                    {*-tSGTw+*,*-pGTw+*,*-kGTw+*,*-sGTw+*,*-tGTw+*,*-kGT+*,
                                    *-tSGT+*,*-tGT+*,*-sGT+*}
QS "C-Labialized"                  {*-tSGTw+*,*-fw+*,*-hw+*,*-dZw+*,*-bw+*,*-dw+*,*-Zw+*,
                                    *-kGTw+*,*-lw+*,*-sw+*,*-mw+*,*-kw+*,*-sGTw+*,*-tw+*,
                                    *-tSw+*,*-tGTw+*,*-gw+*,*-Jw+*,*-vw+*,*-zw+*,*-pw+*,
                                    *-rw+*,*-Sw+*,*-nw+*}
QS "C-Coronal_Fricative"           {*-S+*,*-Z+*,*-Zw+*,*-sw+*,*-sGTw+*,*-zw+*,*-sGT+*,*-s+*,
                                    *-z+*,*-Sw+*}
QS "C-Glottal"                     {*-hw+*,*-h+*}
QS "C-Ejective_plosive"            {*-pGTw+*,*-kGTw+*,*-tGTw+*,*-kGT+*,*-tGT+*}

QS "LL-S"                          {S^*}
QS "LL-Zw"                         {Zw^*}
QS "LL-pau"                        {pau^*}
QS "LL-tSGTw"                      {tSGTw^*}
QS "LL-fw"                         {fw^*}
QS "LL-lw"                         {lw^*}
QS "LL-hw"                         {hw^*}
QS "LL-dZw"                        {dZw^*}
QS "LL-bw"                         {bw^*}
QS "LL-o"                          {o^*}
QS "LL-dw"                         {dw^*}
QS "LL-kGTw"                       {kGTw^*}
QS "LL-wb"                         {wb^*}
QS "LL-sGT"                        {sGT^*}
QS "LL-QQ"                         {QQ^*}
QS "LL-sw"                         {sw^*}
QS "LL-mw"                         {mw^*}
QS "LL-r"                          {r^*}
QS "LL-kw"                         {kw^*}
QS "LL-sGTw"                       {sGTw^*}
QS "LL-h"                          {h^*}
QS "LL-tSw"                        {tSw^*}
QS "LL-dZ"                         {dZ^*}
QS "LL-tGTw"                       {tGTw^*}
QS "LL-Jw"                         {Jw^*}
QS "LL-gw"                         {gw^*}
QS "LL-one"                        {one^*}
QS "LL-J"                          {J^*}
QS "LL-tSGT"                       {tSGT^*}
QS "LL-kGT"                        {kGT^*}
QS "LL-rw"                         {rw^*}
QS "LL-vw"                         {vw^*}
QS "LL-at"                         {at^*}
QS "LL-zw"                         {zw^*}
QS "LL-tS"                         {tS^*}
QS "LL-d"                          {d^*}
QS "LL-a"                          {a^*}
QS "LL-tGT"                        {tGT^*}
QS "LL-b"                          {b^*}
QS "LL-e"                          {e^*}
QS "LL-pw"                         {pw^*}
QS "LL-g"                          {g^*}
QS "LL-f"                          {f^*}
QS "LL-i"                          {i^*}
QS "LL-tw"                         {tw^*}
QS "LL-k"                          {k^*}
QS "LL-j"                          {j^*}
QS "LL-m"                          {m^*}
QS "LL-l"                          {l^*}
QS "LL-pGTw"                       {pGTw^*}
QS "LL-n"                          {n^*}
QS "LL-p"                          {p^*}
QS "LL-s"                          {s^*}
QS "LL-Z"                          {Z^*}
QS "LL-u"                          {u^*}
QS "LL-t"                          {t^*}
QS "LL-w"                          {w^*}
QS "LL-v"                          {v^*}
QS "LL-z"                          {z^*}
QS "LL-Sw"                         {Sw^*}
QS "LL-nw"                         {nw^*}

QS "L-S"                           {*^S-*}
QS "L-Zw"                          {*^Zw-*}
QS "L-pau"                         {*^pau-*}
QS "L-tSGTw"                       {*^tSGTw-*}
QS "L-fw"                          {*^fw-*}
QS "L-lw"                          {*^lw-*}
QS "L-hw"                          {*^hw-*}
QS "L-dZw"                         {*^dZw-*}
QS "L-bw"                          {*^bw-*}
QS "L-o"                           {*^o-*}
QS "L-dw"                          {*^dw-*}
QS "L-kGTw"                        {*^kGTw-*}
QS "L-wb"                          {*^wb-*}
QS "L-sGT"                         {*^sGT-*}
QS "L-QQ"                          {*^QQ-*}
QS "L-sw"                          {*^sw-*}
QS "L-mw"                          {*^mw-*}
QS "L-r"                           {*^r-*}
QS "L-kw"                          {*^kw-*}
QS "L-sGTw"                        {*^sGTw-*}
QS "L-h"                           {*^h-*}
QS "L-tSw"                         {*^tSw-*}
QS "L-dZ"                          {*^dZ-*}
QS "L-tGTw"                        {*^tGTw-*}
QS "L-Jw"                          {*^Jw-*}
QS "L-gw"                          {*^gw-*}
QS "L-one"                         {*^one-*}
QS "L-J"                           {*^J-*}
QS "L-tSGT"                        {*^tSGT-*}
QS "L-kGT"                         {*^kGT-*}
QS "L-rw"                          {*^rw-*}
QS "L-vw"                          {*^vw-*}
QS "L-at"                          {*^at-*}
QS "L-zw"                          {*^zw-*}
QS "L-tS"                          {*^tS-*}
QS "L-d"                           {*^d-*}
QS "L-a"                           {*^a-*}
QS "L-tGT"                         {*^tGT-*}
QS "L-b"                           {*^b-*}
QS "L-e"                           {*^e-*}
QS "L-pw"                          {*^pw-*}
QS "L-g"                           {*^g-*}
QS "L-f"                           {*^f-*}
QS "L-i"                           {*^i-*}
QS "L-tw"                          {*^tw-*}
QS "L-k"                           {*^k-*}
QS "L-j"                           {*^j-*}
QS "L-m"                           {*^m-*}
QS "L-l"                           {*^l-*}
QS "L-pGTw"                        {*^pGTw-*}
QS "L-n"                           {*^n-*}
QS "L-p"                           {*^p-*}
QS "L-s"                           {*^s-*}
QS "L-Z"                           {*^Z-*}
QS "L-u"                           {*^u-*}
QS "L-t"                           {*^t-*}
QS "L-w"                           {*^w-*}
QS "L-v"                           {*^v-*}
QS "L-z"                           {*^z-*}
QS "L-Sw"                          {*^Sw-*}
QS "L-nw"                          {*^nw-*}

QS "C-S"                           {*-S+*}
QS "C-Zw"                          {*-Zw+*}
QS "C-pau"                         {*-pau+*}
QS "C-tSGTw"                       {*-tSGTw+*}
QS "C-fw"                          {*-fw+*}
QS "C-lw"                          {*-lw+*}
QS "C-hw"                          {*-hw+*}
QS "C-dZw"                         {*-dZw+*}
QS "C-bw"                          {*-bw+*}
QS "C-o"                           {*-o+*}
QS "C-dw"                          {*-dw+*}
QS "C-kGTw"                        {*-kGTw+*}
QS "C-wb"                          {*-wb+*}
QS "C-sGT"                         {*-sGT+*}
QS "C-QQ"                          {*-QQ+*}
QS "C-sw"                          {*-sw+*}
QS "C-mw"                          {*-mw+*}
QS "C-r"                           {*-r+*}
QS "C-kw"                          {*-kw+*}
QS "C-sGTw"                        {*-sGTw+*}
QS "C-h"                           {*-h+*}
QS "C-tSw"                         {*-tSw+*}
QS "C-dZ"                          {*-dZ+*}
QS "C-tGTw"                        {*-tGTw+*}
QS "C-Jw"                          {*-Jw+*}
QS "C-gw"                          {*-gw+*}
QS "C-one"                         {*-one+*}
QS "C-J"                           {*-J+*}
QS "C-tSGT"                        {*-tSGT+*}
QS "C-kGT"                         {*-kGT+*}
QS "C-rw"                          {*-rw+*}
QS "C-vw"                          {*-vw+*}
QS "C-at"                          {*-at+*}
QS "C-zw"                          {*-zw+*}

QS "C-tS"                          {*-tS+*}

QS "C-d"                           {*-d+*}

QS "C-a"                           {*-a+*}

QS "C-tGT"                         {*-tGT+*}

QS "C-b"                           {*-b+*}

QS "C-e"                           {*-e+*}

QS "C-pw"                          {*-pw+*}

QS "C-g"                           {*-g+*}

QS "C-f"                           {*-f+*}

QS "C-i"                           {*-i+*}

QS "C-tw"                          {*-tw+*}

QS "C-k"                           {*-k+*}

QS "C-j"                           {*-j+*}

QS "C-m"                           {*-m+*}

QS "C-l"                           {*-l+*}

QS "C-pGTw"                        {*-pGTw+*}

QS "C-n"                           {*-n+*}

QS "C-p"                           {*-p+*}

QS "C-s"                           {*-s+*}

QS "C-Z"                           {*-Z+*}

QS "C-u"                           {*-u+*}

QS "C-t"                           {*-t+*}

QS "C-w"                           {*-w+*}

QS "C-v"                           {*-v+*}

QS "C-z"                           {*-z+*}

QS "C-Sw"                          {*-Sw+*}

QS "C-nw"                          {*-nw+*}


QS "R-S"                           {*+S=*}

QS "R-Zw"                          {*+Zw=*}

QS "R-pau"                         {*+pau=*}

QS "R-tSGTw"                       {*+tSGTw=*}

QS "R-fw"                          {*+fw=*}

QS "R-lw"                          {*+lw=*}

QS "R-hw"                          {*+hw=*}

QS "R-dZw"                         {*+dZw=*}
QS "R-bw"                          {*+bw=*}
QS "R-o"                           {*+o=*}
QS "R-dw"                          {*+dw=*}
QS "R-kGTw"                        {*+kGTw=*}
QS "R-wb"                          {*+wb=*}
QS "R-sGT"                         {*+sGT=*}
QS "R-QQ"                          {*+QQ=*}
QS "R-sw"                          {*+sw=*}
QS "R-mw"                          {*+mw=*}
QS "R-r"                           {*+r=*}
QS "R-kw"                          {*+kw=*}
QS "R-sGTw"                        {*+sGTw=*}
QS "R-h"                           {*+h=*}
QS "R-tSw"                         {*+tSw=*}
QS "R-dZ"                          {*+dZ=*}
QS "R-tGTw"                        {*+tGTw=*}
QS "R-Jw"                          {*+Jw=*}
QS "R-gw"                          {*+gw=*}
QS "R-one"                         {*+one=*}
QS "R-J"                           {*+J=*}
QS "R-tSGT"                        {*+tSGT=*}
QS "R-kGT"                         {*+kGT=*}
QS "R-rw"                          {*+rw=*}
QS "R-vw"                          {*+vw=*}
QS "R-at"                          {*+at=*}
QS "R-zw"                          {*+zw=*}
QS "R-tS"                          {*+tS=*}
QS "R-d"                           {*+d=*}
QS "R-a"                           {*+a=*}
QS "R-tGT"                         {*+tGT=*}
QS "R-b"                           {*+b=*}
QS "R-e"                           {*+e=*}
QS "R-pw"                          {*+pw=*}
QS "R-g"                           {*+g=*}
QS "R-f"                           {*+f=*}
QS "R-i"                           {*+i=*}
QS "R-tw"                          {*+tw=*}
QS "R-k"                           {*+k=*}
QS "R-j"                           {*+j=*}
QS "R-m"                           {*+m=*}
QS "R-l"                           {*+l=*}
QS "R-pGTw"                        {*+pGTw=*}
QS "R-n"                           {*+n=*}
QS "R-p"                           {*+p=*}
QS "R-s"                           {*+s=*}
QS "R-Z"                           {*+Z=*}
QS "R-u"                           {*+u=*}
QS "R-t"                           {*+t=*}
QS "R-w"                           {*+w=*}
QS "R-v"                           {*+v=*}
QS "R-z"                           {*+z=*}
QS "R-Sw"                          {*+Sw=*}
QS "R-nw"                          {*+nw=*}

QS "RR-S"                          {*=S@*}
QS "RR-Zw"                         {*=Zw@*}
QS "RR-pau"                        {*=pau@*}
QS "RR-tSGTw"                      {*=tSGTw@*}
QS "RR-fw"                         {*=fw@*}
QS "RR-lw"                         {*=lw@*}
QS "RR-hw"                         {*=hw@*}
QS "RR-dZw"                        {*=dZw@*}
QS "RR-bw"                         {*=bw@*}
QS "RR-o"                          {*=o@*}
QS "RR-dw"                         {*=dw@*}
QS "RR-kGTw"                       {*=kGTw@*}
QS "RR-wb"                         {*=wb@*}
QS "RR-sGT"                        {*=sGT@*}
QS "RR-QQ"                         {*=QQ@*}
QS "RR-sw"                         {*=sw@*}
QS "RR-mw"                         {*=mw@*}
QS "RR-r"                          {*=r@*}
QS "RR-kw"                         {*=kw@*}
QS "RR-sGTw"                       {*=sGTw@*}
QS "RR-h"                          {*=h@*}
QS "RR-tSw"                        {*=tSw@*}
QS "RR-dZ"                         {*=dZ@*}
QS "RR-tGTw"                       {*=tGTw@*}
QS "RR-Jw"                         {*=Jw@*}
QS "RR-gw"                         {*=gw@*}
QS "RR-one"                        {*=one@*}
QS "RR-J"                          {*=J@*}
QS "RR-tSGT"                       {*=tSGT@*}
QS "RR-kGT"                        {*=kGT@*}
QS "RR-rw"                         {*=rw@*}
QS "RR-vw"                         {*=vw@*}
QS "RR-at"                         {*=at@*}
QS "RR-zw"                         {*=zw@*}
QS "RR-tS"                         {*=tS@*}
QS "RR-d"                          {*=d@*}
QS "RR-a"                          {*=a@*}
QS "RR-tGT"                        {*=tGT@*}
QS "RR-b"                          {*=b@*}
QS "RR-e"                          {*=e@*}
QS "RR-pw"                         {*=pw@*}
QS "RR-g"                          {*=g@*}
QS "RR-f"                          {*=f@*}
QS "RR-i"                          {*=i@*}
QS "RR-tw"                         {*=tw@*}
QS "RR-k"                          {*=k@*}
QS "RR-j"                          {*=j@*}
QS "RR-m"                          {*=m@*}
QS "RR-l"                          {*=l@*}
QS "RR-pGTw"                       {*=pGTw@*}
QS "RR-n"                          {*=n@*}
QS "RR-p"                          {*=p@*}
QS "RR-s"                          {*=s@*}
QS "RR-Z"                          {*=Z@*}
QS "RR-u"                          {*=u@*}
QS "RR-t"                          {*=t@*}
QS "RR-w"                          {*=w@*}
QS "RR-v"                          {*=v@*}
QS "RR-z"                          {*=z@*}
QS "RR-Sw"                         {*=Sw@*}
QS "RR-nw"                         {*=nw@*}

QS "C--Syl_Front_Vowel"            {*|a/C/*,*|e/C/*,*|i/C/*}
QS "C--Syl_High_Vowel"             {*|a/C/*}
QS "C--Syl_Vowel"                  {*|at/C/*,*|a/C/*,*|e/C/*,*|i/C/*,*|o/C/*,*|u/C/*,
                                    *|one/C/*}
QS "C--Syl_Mid_Vowel"              {*|at/C/*,*|e/C/*,*|o/C/*}
QS "C--Syl_Unrounded"              {*|at/C/*,*|a/C/*,*|e/C/*,*|i/C/*,*|one/C/*}
QS "C--Syl_Central_Vowel"          {*|at/C/*,*|one/C/*}
QS "C--Syl_Back_Vowel"             {*|o/C/*,*|u/C/*}
QS "C--Syl_Rounded"                {*|o/C/*,*|u/C/*}
```

Appendix D

Custom Question Set for Turkish

```
QS "LL-Vowel"               {i^*,e^*,y^*,yy^*,two^*,twotwo^*,u^*,o^*,one^*,a^*,ii^*,ee^*,uu^*,
                             oo^*,oneone^*,aa^*}
QS "LL-Consonant"           {p^*,b^*,t^*,d^*,c^*,gj^*,k^*,g^*,f^*,v^*,s^*,z^*,S^*,Z^*,h^*,G^*,
                             tS^*,dZ^*,m^*,n^*,l^*,five^*,r^*,j^*,w^*,N^*}
QS "LL-Stop"                {p^*,b^*,t^*,d^*,c^*,gj^*,k^*,g^*,tS^*,dZ^*}
QS "LL-Nasal"               {m^*,n^*,N^*}
QS "LL-Fricative"           {f^*,v^*,s^*,z^*,S^*,Z^*,h^*,G^*}
QS "LL-Liquid"              {l^*,five^*,r^*}
QS "LL-Front"               {i^*,ii^*,e^*,ee^*,y^*,yy^*,two^*,twotwo^*,p^*,b^*,m^*,f^*,v^*,m^*,
                             w^*}
QS "LL-Central"             {d^*,dZ^*,l^*,five^*,n^*,r^*,s^*,S^*,t^*,z^*,Z^*}
QS "LL-Back"                {u^*,uu^*,o^*,oo^*,one^*,oneone^*,a^*,aa^*,c^*,gj^*,g^*,G^*,h^*,j^*,
                             k^*}
QS "LL-Front_Vowel"         {i^*,ii^*,e^*,ee^*,y^*,yy^*,two^*,twotwo^*}
QS "LL-Back_Vowel"          {u^*,uu^*,o^*,oo^*,one^*,oneone^*,a^*,aa^*}
QS "LL-Long_Vowel"          {ii^*,ee^*,uu^*,oo^*,yy^*,oneone^*,aa^*,twotwo^*}
QS "LL-Short_Vowel"         {i^*,e^*,y^*,two^*,u^*,o^*,one^*,a^*}
QS "LL-Front_Start_Vowel"   {i^*,ii^*,e^*,ee^*,y^*,yy^*,two^*,twotwo^*}
QS "LL-High_Vowel"          {i^*,ii^*,y^*,yy^*,u^*,uu^*,one^*,oneone^*}
QS "LL-Medium_Vowel"        {e^*,ee^*,two^*,twotwo^*,o^*,oo^*}
QS "LL-Low_Vowel"           {a^*,aa^*}
QS "LL-Rounded_Vowel"       {w^*}
QS "LL-Unrounded_Vowel"     {i^*,ii^*,e^*,ee^*,y^*,yy^*,two^*,twotwo^*,u^*,uu^*,o^*,oo^*,one^*,
                             oneone^*,a^*,aa^*}
QS "LL-IVowel"              {i^*,ii^*,y^*,yy^*}
QS "LL-EVowel"              {e^*,ee^*}
QS "LL-AVowel"              {a^*,aa^*}
QS "LL-OVowel"              {o^*,oo^*,two^*,twotwo^*}
QS "LL-UVowel"              {u^*,uu^*,one^*,oneone^*}
QS "LL-Unvoiced_Consonant"  {p^*,t^*,c^*,k^*,f^*,s^*,S^*,h^*,tS^*}
QS "LL-Voiced_Consonant"    {b^*,d^*,gj^*,g^*,v^*,z^*,Z^*,G^*,dZ^*,m^*,n^*,l^*,five^*,r^*,j^*,
                             w^*,N^*}
QS "LL-Front_Consonant"     {p^*,b^*,m^*,f^*,v^*,m^*,w^*}
QS "LL-Central_Consonant"   {d^*,dZ^*,l^*,five^*,n^*,r^*,s^*,S^*,t^*,z^*,Z^*}
QS "LL-Back_Consonant"      {c^*,gj^*,g^*,G^*,h^*,j^*,k^*}
QS "LL-Coronal_Consonant"   {t^*,d^*,s^*,z^*,S^*,Z^*,tS^*,dZ^*,n^*,r^*,N^*,l^*,five^*}
QS "LL-Non_Coronal"         {p^*,b^*,f^*,c^*,gj^*,k^*,g^*,f^*,v^*,h^*,G^*,m^*,j^*,w^*}
QS "LL-Anterior_Consonant"  {b^*,d^*,f^*,l^*,five^*,m^*,n^*,N^*,p^*,s^*,t^*,v^*,w^*,z^*}
QS "LL-Non_Anterior"        {tS^*,g^*,h^*,Z^*,c^*,k^*,r^*,S^*,y^*,yy^*,dZ^*}
QS "LL-Continuent"          {f^*,v^*,s^*,z^*,S^*,Z^*,h^*,G^*,l^*,five^*,r^*,j^*,w^*}
QS "LL-No_Continuent"       {p^*,b^*,t^*,d^*,c^*,gj^*,k^*,g^*,tS^*,dZ^*,m^*,n^*,N^*}
QS "LL-Glide"               {h^*,l^*,five^*,r^*,y^*,yy^*,w^*}
QS "LL-Voiced_Stop"         {b^*,d^*,g^*,gj^*,dZ^*}
QS "LL-Unvoiced_Stop"       {p^*,t^*,k^*,c^*,tS^*}
QS "LL-Front_Stop"          {b^*,p^*}
QS "LL-Central_Stop"        {d^*,t^*,tS^*,dZ^*}
QS "LL-Back_Stop"           {g^*,k^*,c^*}
QS "LL-Voiced_Fricative"    {v^*,z^*,Z^*,G^*}
QS "LL-Unvoiced_Fricative"  {f^*,s^*,S^*,h^*}
QS "LL-Front_Fricative"     {f^*,v^*}
QS "LL-Central_Fricative"   {s^*,z^*}
QS "LL-Back_Fricative"      {S^*,Z^*}
QS "LL-Affricate_Consonant" {tS^*,dZ^*}
QS "LL-silences"            {pau^*,h#^*,brth^*,wb^*}
QS "LL-i"                   {i^*}
QS "LL-ii"                  {ii^*}
QS "LL-e"                   {e^*}
QS "LL-ee"                  {ee^*}
QS "LL-y"                   {y^*}
QS "LL-yy"                  {yy^*}
QS "LL-two"                 {two^*}
QS "LL-twotwo"              {twotwo^*}
QS "LL-u"                   {u^*}
QS "LL-uu"                  {uu^*}
QS "LL-o"                   {o^*}
QS "LL-oo"                  {oo^*}
QS "LL-one"                 {one^*}
QS "LL-oneone"              {oneone^*}
QS "LL-a"                   {a^*}
QS "LL-aa"                  {aa^*}
QS "LL-p"                   {p^*}
QS "LL-b"                   {b^*}
QS "LL-t"                   {t^*}
QS "LL-d"                   {d^*}
QS "LL-c"                   {c^*}
QS "LL-gj"                  {gj^*}
QS "LL-k"                   {k^*}
QS "LL-g"                   {g^*}
QS "LL-f"                   {f^*}
QS "LL-v"                   {v^*}
QS "LL-s"                   {s^*}
QS "LL-z"                   {z^*}
QS "LL-S"                   {S^*}
QS "LL-Z"                   {Z^*}
QS "LL-h"                   {h^*}
QS "LL-G"                   {G^*}
QS "LL-tS"                  {tS^*}
QS "LL-dZ"                  {dZ^*}
QS "LL-m"                   {m^*}
QS "LL-n"                   {n^*}
QS "LL-l"                   {l^*}
QS "LL-five"                {five^*}
QS "LL-r"                   {r^*}
QS "LL-j"                   {j^*}
QS "LL-w"                   {w^*}
QS "LL-N"                   {N^*}
QS "LL-pau"                 {pau^*}
QS "LL-brth"                {brth^*}
QS "LL-wb"                  {wb^*}

QS "L-Vowel"                {*^i-*,*^e-*,*^y-*,*^yy-*,*^two-*,*^twotwo-*,*^u-*,*^o-*,*^one-*,
                             *^a-*,*^ii-*,*^ee-*,*^uu-*,*^oo-*,*^oneone-*,*^aa-*}
QS "L-Consonant"            {*^p-*,*^b-*,*^t-*,*^d-*,*^c-*,*^gj-*,*^k-*,*^g-*,*^f-*,*^v-*,
                             *^s-*,*^z-*,*^S-*,*^Z-*,*^h-*,*^G-*,*^tS-*,*^dZ-*,*^m-*,*^n-*,
                             *^l-*,*^five-*,*^r-*,*^j-*,*^w-*,*^N-*}
QS "L-Stop"                 {*^p-*,*^b-*,*^t-*,*^d-*,*^c-*,*^gj-*,*^k-*,*^g-*,*^tS-*,*^dZ-*}
QS "L-Nasal"                {*^m-*,*^n-*,*^N-*}
QS "L-Fricative"            {*^f-*,*^v-*,*^s-*,*^z-*,*^S-*,*^Z-*,*^h-*,*^G-*}
QS "L-Liquid"               {*^l-*,*^five-*,*^r-*}
QS "L-Front"                {*^i-*,*^ii-*,*^e-*,*^ee-*,*^y-*,*^yy-*,*^two-*,^twotwo-*,*^p-*,
                             *^b-*,*^m-*,*^f-*,*^v-*,*^m-*,*^w-*}
QS "L-Central"              {*^d-*,*^dZ-*,*^l-*,*^five-*,*^n-*,*^r-*,*^s-*,*^S-*,*^t-*,*^z-*,
                             *^Z-*}
QS "L-Back"                 {*^u-*,*^uu-*,*^o-*,*^oo-*,*^one-*,*^oneone-*,*^a-*,*^aa-*,*^c-*,
                             *^gj-*,*^g-*,*^G-*,*^h-*,*^j-*,*^k-*}
QS "L-Front_Vowel"          {*^i-*,*^ii-*,*^e-*,*^ee-*,*^y-*,*^yy-*,*^two-*,*^twotwo-*}
QS "L-Back_Vowel"           {*^u-*,*^uu-*,*^o-*,*^oo-*,*^one-*,*^oneone-*,*^a-*,*^aa-*}
QS "L-Long_Vowel"           {*^ii-*,*^ee-*,*^uu-*,*^oo-*,*^yy-*,*^oneone-*,*^aa-*,*^twotwo-*}
QS "L-Short_Vowel"          {*^i-*,*^e-*,*^y-*,*^two-*,*^u-*,*^o-*,*^one-*,*^a-*}
QS "L-Front_Start_Vowel"    {*^i-*,*^ii-*,*^e-*,*^ee-*,*^y-*,*^yy-*,*^two-*,*^twotwo-*}
QS "L-High_Vowel"           {*^i-*,*^ii-*,*^y-*,*^yy-*,*^u-*,*^uu-*,*^one-*,*^oneone-*}
QS "L-Medium_Vowel"         {*^e-*,*^ee-*,*^two-*,*^twotwo-*,*^o-*,*^oo-*}
QS "L-Low_Vowel"            {*^a-*,*^aa-*}
QS "L-Rounded_Vowel"        {*^w-*}
QS "L-Unrounded_Vowel"      {*^i-*,*^ii-*,*^e-*,*^ee-*,*^y-*,*^yy-*,*^two-*,*^twotwo-*,*^u-*,
                             *^uu-*,*^o-*,*^oo-*,*^one-*,*^oneone-*,*^a-*,*^aa-*}
QS "L-IVowel"               {*^i-*,*^ii-*,*^y-*,*^yy-*}
QS "L-EVowel"               {*^e-*,*^ee-*}
QS "L-AVowel"               {*^a-*,*^aa-*}
QS "L-OVowel"               {*^o-*,*^oo-*,*^two-*,*^twotwo-*}
QS "L-UVowel"               {*^u-*,*^uu-*,*^one-*,*^oneone-*}
QS "L-Unvoiced_Consonant"   {*^p-*,*^t-*,*^c-*,*^k-*,*^f-*,*^s-*,*^S-*,*^h-*,*^tS-*}
QS "L-Voiced_Consonant"     {*^b-*,*^d-*,*^gj-*,*^g-*,*^v-*,*^z-*,*^Z-*,*^G-*,*^dZ-*,*^m-*,
                             *^n-*,*^l-*,*^five-*,*^r-*,*^j-*,*^w-*,*^N-*}
```

```
QS "L-Front_Consonant"     {*^p-*,*^b-*,*^m-*,*^f-*,*^v-*,*^m-*,*^w-*}
QS "L-Central_Consonant"   {*^d-*,*^dZ-*,*^l-*,*^five-*,*^n-*,*^r-*,*^s-*,*^S-*,*^t-*,*^z-*,
                            *^Z-*}
QS "L-Back_Consonant"      {*^c-*,*^gj-*,*^g-*,*^G-*,*^h-*,*^j-*,*^k-*}
QS "L-Coronal_Consonant"   {*^t-*,*^d-*,*^s-*,*^z-*,*^S-*,*^Z-*,*^tS-*,*^dZ-*,*^n-*,*^r-*,
                            *^N-*,*^l-*,*^five-*}
QS "L-Non_Coronal"         {*^p-*,*^b-*,*^f-*,*^c-*,*^gj-*,*^k-*,*^g-*,*^f-*,*^v-*,*^h-*,
                            *^G-*,*^m-*,*^j-*,*^w-*}
QS "L-Anterior_Consonant"  {*^b-*,*^d-*,*^f-*,*^l-*,*^five-*,*^m-*,*^n-*,*^N-*,*^p-*,*^s-*,
                            *^t-*,*^v-*,*^w-*,*^z-*}
QS "L-Non_Anterior"        {*^tS-*,*^g-*,*^h-*,*^Z-*,*^c-*,*^k-*,*^r-*,*^S-*,*^y-*,*^yy-*,
                            *^dZ-*}
QS "L-Continuent"          {*^f-*,*^v-*,*^s-*,*^z-*,*^S-*,*^Z-*,*^h-*,*^G-*,*^l-*,*^five-*,
                            *^r-*,*^j-*,*^w-*}
QS "L-No_Continuent"       {*^p-*,*^b-*,*^t-*,*^d-*,*^c-*,*^gj-*,*^k-*,*^g-*,*^tS-*,*^dZ-*,
                            *^m-*,*^n-*,*^N-*}
QS "L-Glide"               {*^h-*,*^l-*,*^five-*,*^r-*,*^y-*,*^yy-*,*^w-*}
QS "L-Voiced_Stop"         {*^b-*,*^d-*,*^g-*,*^gj-*,*^dZ-*}
QS "L-Unvoiced_Stop"       {*^p-*,*^t-*,*^k-*,*^c-*,*^tS-*}
QS "L-Front_Stop"          {*^b-*,*^p-*}
QS "L-Central_Stop"        {*^d-*,*^t-*,*^tS-*,*^dZ-*}
QS "L-Back_Stop"           {*^g-*,*^k-*,*^c-*}
QS "L-Voiced_Fricative"    {*^v-*,*^z-*,*^Z-*,*^G-*}
QS "L-Unvoiced_Fricative"  {*^f-*,*^s-*,*^S-*,*^h-*}
QS "L-Front_Fricative"     {*^f-*,*^v-*}
QS "L-Central_Fricative"   {*^s-*,*^z-*}
QS "L-Back_Fricative"      {*^S-*,*^Z-*}
QS "L-Affricate_Consonant" {*^tS-*,*^dZ-*}
QS "L-silences"            {*^pau-*,*^h#-*,*^brth-*,*^wb-*}
QS "L-i"                   {*^i-*}
QS "L-ii"                  {*^ii-*}
QS "L-e"                   {*^e-*}
QS "L-ee"                  {*^ee-*}
QS "L-y"                   {*^y-*}
QS "L-yy"                  {*^yy-*}
QS "L-two"                 {*^two-*}
QS "L-twotwo"              {*^twotwo-*}
QS "L-u"                    {*^u-*}

QS "L-uu"                   {*^uu-*}

QS "L-o"                    {*^o-*}

QS "L-oo"                   {*^oo-*}

QS "L-one"                  {*^one-*}

QS "L-oneone"               {*^oneone-*}

QS "L-a"                    {*^a-*}

QS "L-aa"                   {*^aa-*}

QS "L-p"                    {*^p-*}

QS "L-b"                    {*^b-*}

QS "L-t"                    {*^t-*}

QS "L-d"                    {*^d-*}

QS "L-c"                    {*^c-*}

QS "L-gj"                   {*^gj-*}

QS "L-k"                    {*^k-*}

QS "L-g"                    {*^g-*}

QS "L-f"                    {*^f-*}

QS "L-v"                    {*^v-*}

QS "L-s"                    {*^s-*}

QS "L-z"                    {*^z-*}

QS "L-S"                    {*^S-*}

QS "L-Z"                    {*^Z-*}

QS "L-h"                    {*^h-*}

QS "L-G"                    {*^G-*}

QS "L-tS"                   {*^tS-*}

QS "L-dZ"                   {*^dZ-*}

QS "L-m"                    {*^m-*}

QS "L-n"                    {*^n-*}

QS "L-l"                    {*^l-*}

QS "L-five"                 {*^five-*}

QS "L-r"                    {*^r-*}

QS "L-j"                    {*^j-*}

QS "L-w"                    {*^w-*}

QS "L-N"                    {*^N-*}

QS "L-pau"                  {*^pau-*}

QS "L-brth"                 {*^brth-*}

QS "L-wb"                   {*^wb-*}
QS "C-Vowel"                {*-i+*,*-e+*,*-y+*,*-yy+*,*-two+*,*-twotwo+*,*-u+*,*-o+*,
                             *-one+*,*-a+*,*-ii+*,*-ee+*,*-uu+*,*-oo+*,*-oneone+*,*-aa+*}
QS "C-Consonant"            {*-p+*,*-b+*,*-t+*,*-d+*,*-c+*,*-gj+*,*-k+*,*-g+*,*-f+*,*-v+*,
                             *-s+*,*-z+*,*-S+*,*-Z+*,*-h+*,*-G+*,*-tS+*,*-dZ+*,*-m+*,*-n+*,
                             *-l+*,*-five+*,*-r+*,*-j+*,*-w+*,*-N+*}
QS "C-Stop"                 {*-p+*,*-b+*,*-t+*,*-d+*,*-c+*,*-gj+*,*-k+*,*-g+*,*-tS+*,
                             *-dZ+*}
QS "C-Nasal"                {*-m+*,*-n+*,*-N+*}
QS "C-Fricative"            {*-f+*,*-v+*,*-s+*,*-z+*,*-S+*,*-Z+*,*-h+*,*-G+*}
QS "C-Liquid"               {*-l+*,*-five+*,*-r+*}
QS "C-Front"                {*-i+*,*-ii+*,*-e+*,*-ee+*,*-y+*,*-yy+*,*-two+*,*-twotwo+*,
                             *-p+*,*-b+*,*-m+*,*-f+*,*-v+*,*-m+*,*-w+*}
QS "C-Central"              {*-d+*,*-dZ+*,*-l+*,*-five+*,*-n+*,*-r+*,*-s+*,*-S+*,*-t+*,
                             *-z+*,*-Z+*}
QS "C-Back"                 {*-u+*,*-uu+*,*-o+*,*-oo+*,*-one+*,*-oneone+*,*-a+*,*-aa+*,
                             *-c+*,*-gj+*,*-g+*,*-G+*,*-h+*,*-j+*,*-k+*}
QS "C-Front_Vowel"          {*-i+*,*-ii+*,*-e+*,*-ee+*,*-y+*,*-yy+*,*-two+*,*-twotwo+*}
QS "C-Back_Vowel"           {*-u+*,*-uu+*,*-o+*,*-oo+*,*-one+*,*-oneone+*,*-a+*,*-aa+*}
QS "C-Long_Vowel"           {*-ii+*,*-ee+*,*-uu+*,*-oo+*,*-yy+*,*-oneone+*,*-aa+*,
                             *-twotwo+*}
QS "C-Short_Vowel"          {*-i+*,*-e+*,*-y+*,*-two+*,*-u+*,*-o+*,*-one+*,*-a+*}
QS "C-Front_Start_Vowel"    {*-i+*,*-ii+*,*-e+*,*-ee+*,*-y+*,*-yy+*,*-two+*,*-twotwo+*}
QS "C-High_Vowel"           {*-i+*,*-ii+*,*-y+*,*-yy+*,*-u+*,*-uu+*,*-one+*,*-oneone+*}
QS "C-Medium_Vowel"         {*-e+*,*-ee+*,*-two+*,*-twotwo+*,*-o+*,*-oo+*}
QS "C-Low_Vowel"            {*-a+*,*-aa+*}
QS "C-Rounded_Vowel"        {*-w+*}
QS "C-Unrounded_Vowel"      {*-i+*,*-ii+*,*-e+*,*-ee+*,*-y+*,*-yy+*,*-two+*,*-twotwo+*,
                             *-u+*,*-uu+*,*-o+*,*-oo+*,*-one+*,*-oneone+*,*-a+*,*-aa+*}
QS "C-IVowel"               {*-i+*,*-ii+*,*-y+*,*-yy+*}
QS "C-EVowel"               {*-e+*,*-ee+*}
QS "C-AVowel"               {*-a+*,*-aa+*}
QS "C-OVowel"               {*-o+*,*-oo+*,*-two+*,*-twotwo+*}
QS "C-UVowel"               {*-u+*,*-uu+*,*-one+*,*-oneone+*}
QS "C-Unvoiced_Consonant"   {*-p+*,*-t+*,*-c+*,*-k+*,*-f+*,*-s+*,*-S+*,*-h+*,*-tS+*}
QS "C-Voiced_Consonant"     {*-b+*,*-d+*,*-gj+*,*-g+*,*-v+*,*-z+*,*-Z+*,*-G+*,*-dZ+*,*-m+*,
                             *-n+*,*-l+*,*-five+*,*-r+*,*-j+*,*-w+*,*-N+*}
```

```
QS "C-Front_Consonant"     {*-p+*,*-b+*,*-m+*,*-f+*,*-v+*,*-m+*,*-w+*}
QS "C-Central_Consonant"   {*-d+*,*-dZ+*,*-l+*,*-five+*,*-n+*,*-r+*,*-s+*,*-S+*,*-t+*,
                            *-z+*,*-Z+*}
QS "C-Back_Consonant"      {*-c+*,*-gj+*,*-g+*,*-G+*,*-h+*,*-j+*,*-k+*}
QS "C-Coronal_Consonant"   {*-t+*,*-d+*,*-s+*,*-z+*,*-S+*,*-Z+*,*-tS+*,*-dZ+*,*-n+*,*-r+*,
                            *-N+*,*-l+*,*-five+*}
QS "C-Non_Coronal"         {*-p+*,*-b+*,*-f+*,*-c+*,*-gj+*,*-k+*,*-g+*,*-f+*,*-v+*,*-h+*,
                            *-G+*,*-m+*,*-j+*,*-w+*}
QS "C-Anterior_Consonant"  {*-b+*,*-d+*,*-f+*,*-l+*,*-five+*,*-m+*,*-n+*,*-N+*,*-p+*,
                            *-s+*,*-t+*,*-v+*,*-w+*,*-z+*}
QS "C-Non_Anterior"        {*-tS+*,*-g+*,*-h+*,*-Z+*,*-c+*,*-k+*,*-r+*,*-S+*,*-y+*,*-yy+*,
                            *-dZ+*}
QS "C-Continuent"          {*-f+*,*-v+*,*-s+*,*-z+*,*-S+*,*-Z+*,*-h+*,*-G+*,*-l+*,*-five+*,
                            *-r+*,*-j+*,*-w+*}
QS "C-No_Continuent"       {*-p+*,*-b+*,*-t+*,*-d+*,*-c+*,*-gj+*,*-k+*,*-g+*,*-tS+*,*-dZ+*,
                            *-m+*,*-n+*,*-N+*}
QS "C-Glide"               {*-h+*,*-l+*,*-five+*,*-r+*,*-y+*,*-yy+*,*-w+*}
QS "C-Voiced_Stop"         {*-b+*,*-d+*,*-g+*,*-gj+*,*-dZ+*}
QS "C-Unvoiced_Stop"       {*-p+*,*-t+*,*-k+*,*-c+*,*-tS+*}
QS "C-Front_Stop"          {*-b+*,*-p+*}
QS "C-Central_Stop"        {*-d+*,*-t+*,*-tS+*,*-dZ+*}
QS "C-Back_Stop"           {*-g+*,*-k+*,*-c+*}
QS "C-Voiced_Fricative"    {*-v+*,*-z+*,*-Z+*,*-G+*}
QS "C-Unvoiced_Fricative"  {*-f+*,*-s+*,*-S+*,*-h+*}
QS "C-Front_Fricative"     {*-f+*,*-v+*}
QS "C-Central_Fricative"   {*-s+*,*-z+*}
QS "C-Back_Fricative"      {*-S+*,*-Z+*}
QS "C-Affricate_Consonant" {*-tS+*,*-dZ+*}
QS "C-silences"            {*-pau+*,*-h#+*,*-brth+*,*-wb+*}
QS "C-i"                   {*-i+*}
QS "C-ii"                  {*-ii+*}
QS "C-e"                   {*-e+*}
QS "C-ee"                  {*-ee+*}
QS "C-y"                   {*-y+*}
QS "C-yy"                  {*-yy+*}
QS "C-two"                 {*-two+*}
QS "C-twotwo"              {*-twotwo+*}
QS "C-u"                    {*-u+*}
QS "C-uu"                   {*-uu+*}
QS "C-o"                    {*-o+*}
QS "C-oo"                   {*-oo+*}
QS "C-one"                  {*-one+*}
QS "C-oneone"               {*-oneone+*}
QS "C-a"                    {*-a+*}
QS "C-aa"                   {*-aa+*}
QS "C-p"                    {*-p+*}
QS "C-b"                    {*-b+*}
QS "C-t"                    {*-t+*}
QS "C-d"                    {*-d+*}
QS "C-c"                    {*-c+*}
QS "C-gj"                   {*-gj+*}
QS "C-k"                    {*-k+*}
QS "C-g"                    {*-g+*}
QS "C-f"                    {*-f+*}
QS "C-v"                    {*-v+*}
QS "C-s"                    {*-s+*}
QS "C-z"                    {*-z+*}
QS "C-S"                    {*-S+*}
QS "C-Z"                    {*-Z+*}
QS "C-h"                    {*-h+*}
QS "C-G"                    {*-G+*}
QS "C-tS"                   {*-tS+*}
QS "C-dZ"                   {*-dZ+*}
QS "C-m"                    {*-m+*}
QS "C-n"                    {*-n+*}
QS "C-l"                    {*-l+*}
QS "C-five"                 {*-five+*}
QS "C-r"                    {*-r+*}
QS "C-j"                    {*-j+*}
QS "C-w"                    {*-w+*}
QS "C-N"                    {*-N+*}
QS "C-pau"                  {*-pau+*}
QS "C-brth"                 {*-brth+*}
QS "C-wb"                   {*-wb+*}
QS "R-Vowel"                {*+i=*,*+e=*,*+y=*,*+yy=*,*+two=*,*+twotwo=*,*+u=*,*+o=*,
                             *+one=*,*+a=*,*+ii=*,*+ee=*,*+uu=*,*+oo=*,*+oneone=*,*+aa=*}
QS "R-Consonant"            {*+p=*,*+b=*,*+t=*,*+d=*,*+c=*,*+gj=*,*+k=*,*+g=*,*+f=*,*+v=*,
                             *+s=*,*+z=*,*+S=*,*+Z=*,*+h=*,*+G=*,*+tS=*,*+dZ=*,*+m=*,
                             *+n=*,*+l=*,*+five=*,*+r=*,*+j=*,*+w=*,*+N=*}
QS "R-Stop"                 {*+p=*,*+b=*,*+t=*,*+d=*,*+c=*,*+gj=*,*+k=*,*+g=*,*+tS=*,
                             *+dZ=*}
QS "R-Nasal"                {*+m=*,*+n=*,*+N=*}
QS "R-Fricative"            {*+f=*,*+v=*,*+s=*,*+z=*,*+S=*,*+Z=*,*+h=*,*+G=*}
QS "R-Liquid"               {*+l=*,*+five=*,*+r=*}
QS "R-Front"                {*+i=*,*+ii=*,*+e=*,*+ee=*,*+y=*,*+yy=*,*+two=*,*+twotwo=*,
                             *+p=*,*+b=*,*+m=*,*+f=*,*+v=*,*+m=*,*+w=*}
QS "R-Central"              {*+d=*,*+dZ=*,*+l=*,*+five=*,*+n=*,*+r=*,*+s=*,*+S=*,*+t=*,
                             *+z=*,*+Z=*}
QS "R-Back"                 {*+u=*,*+uu=*,*+o=*,*+oo=*,*+one=*,*+oneone=*,*+a=*,*+aa=*,
                             *+c=*,*+gj=*,*+g=*,*+G=*,*+h=*,*+j=*,*+k=*}
QS "R-Front_Vowel"          {*+i=*,*+ii=*,*+e=*,*+ee=*,*+y=*,*+yy=*,*+two=*,*+twotwo=*}
QS "R-Back_Vowel"           {*+u=*,*+uu=*,*+o=*,*+oo=*,*+one=*,*+oneone=*,*+a=*,*+aa=*}
QS "R-Long_Vowel"           {*+ii=*,*+ee=*,*+uu=*,*+oo=*,*+yy=*,*+oneone=*,*+aa=*,
                             *+twotwo=*}
QS "R-Short_Vowel"          {*+i=*,*+e=*,*+y=*,*+two=*,*+u=*,*+o=*,*+one=*,*+a=*}
QS "R-Front_Start_Vowel"    {*+i=*,*+ii=*,*+e=*,*+ee=*,*+y=*,*+yy=*,*+two=*,*+twotwo=*}
QS "R-High_Vowel"           {*+i=*,*+ii=*,*+y=*,*+yy=*,*+u=*,*+uu=*,*+one=*,*+oneone=*}
QS "R-Medium_Vowel"         {*+e=*,*+ee=*,*+two=*,*+twotwo=*,*+o=*,*+oo=*}
QS "R-Low_Vowel"            {*+a=*,*+aa=*}
QS "R-Rounded_Vowel"        {*+w=*}
QS "R-Unrounded_Vowel"      {*+i=*,*+ii=*,*+e=*,*+ee=*,*+y=*,*+yy=*,*+two=*,*+twotwo=*,
                             *+u=*,*+uu=*,
                             *+o=*,*+oo=*,*+one=*,*+oneone=*,*+a=*,*+aa=*}
QS "R-IVowel"               {*+i=*,*+ii=*,*+y=*,*+yy=*}
QS "R-EVowel"               {*+e=*,*+ee=*}
QS "R-AVowel"               {*+a=*,*+aa=*}
QS "R-OVowel"               {*+o=*,*+oo=*,*+two=*,*+twotwo=*}
QS "R-UVowel"               {*+u=*,*+uu=*,*+one=*,*+oneone=*}
QS "R-Unvoiced_Consonant"   {*+p=*,*+t=*,*+c=*,*+k=*,*+f=*,*+s=*,*+S=*,*+h=*,*+tS=*}
QS "R-Voiced_Consonant"     {*+b=*,*+d=*,*+gj=*,*+g=*,*+v=*,*+z=*,*+Z=*,*+G=*,*+dZ=*,
                            *+m=*,*+n=*,*+l=*,*+five=*,*+r=*,*+j=*,*+w=*,*+N=*}
QS "R-Front_Consonant"     {*+p=*,*+b=*,*+m=*,*+f=*,*+v=*,*+m=*,*+w=*}
QS "R-Central_Consonant"   {*+d=*,*+dZ=*,*+l=*,*+five=*,*+n=*,*+r=*,*+s=*,*+S=*,*+t=*,
                            *+z=*,*+Z=*}
QS "R-Back_Consonant"      {*+c=*,*+gj=*,*+g=*,*+G=*,*+h=*,*+j=*,*+k=*}
QS "R-Coronal_Consonant"   {*+t=*,*+d=*,*+s=*,*+z=*,*+S=*,*+Z=*,*+tS=*,*+dZ=*,*+n=*,
                            *+r=*,*+N=*,*+l=*,*+five=*}
QS "R-Non_Coronal"         {*+p=*,*+b=*,*+f=*,*+c=*,*+gj=*,*+k=*,*+g=*,*+f=*,*+v=*,
                            *+h=*,*+G=*,*+m=*,*+j=*,*+w=*}
QS "R-Anterior_Consonant"  {*+b=*,*+d=*,*+f=*,*+l=*,*+five=*,*+m=*,*+n=*,*+N=*,*+p=*,
                            *+s=*,*+t=*,*+v=*,*+w=*,*+z=*}
QS "R-Non_Anterior"        {*+tS=*,*+g=*,*+h=*,*+Z=*,*+c=*,*+k=*,*+r=*,*+S=*,*+y=*,
                            *+yy=*,*+dZ=*}
QS "R-Continuent"          {*+f=*,*+v=*,*+s=*,*+z=*,*+S=*,*+Z=*,*+h=*,*+G=*,*+l=*,
                            *+five=*,*+r=*,*+j=*,*+w=*}
QS "R-No_Continuent"       {*+p=*,*+b=*,*+t=*,*+d=*,*+c=*,*+gj=*,*+k=*,*+g=*,*+tS=*,
                            *+dZ=*,*+m=*,*+n=*,*+N=*}
QS "R-Glide"               {*+h=*,*+l=*,*+five=*,*+r=*,*+y=*,*+yy=*,*+w=*}
QS "R-Voiced_Stop"         {*+b=*,*+d=*,*+g=*,*+gj=*,*+dZ=*}
QS "R-Unvoiced_Stop"       {*+p=*,*+t=*,*+k=*,*+c=*,*+tS=*}
QS "R-Front_Stop"          {*+b=*,*+p=*}
QS "R-Central_Stop"        {*+d=*,*+t=*,*+tS=*,*+dZ=*}
QS "R-Back_Stop"           {*+g=*,*+k=*,*+c=*}
QS "R-Voiced_Fricative"    {*+v=*,*+z=*,*+Z=*,*+G=*}
QS "R-Unvoiced_Fricative"  {*+f=*,*+s=*,*+S=*,*+h=*}
QS "R-Front_Fricative"     {*+f=*,*+v=*}
QS "R-Central_Fricative"   {*+s=*,*+z=*}
QS "R-Back_Fricative"      {*+S=*,*+Z=*}
QS "R-Affricate_Consonant" {*+tS=*,*+dZ=*}
QS "R-silences"            {*+pau=*,*+h#=*,*+brth=*,*+wb=*}
QS "R-i"                   {*+i=*}
QS "R-ii"                  {*+ii=*}
QS "R-e"                   {*+e=*}
QS "R-ee"                  {*+ee=*}
QS "R-y"                   {*+y=*}
QS "R-yy"                  {*+yy=*}
QS "R-two"                 {*+two=*}
QS "R-twotwo"               {*+twotwo=*}
QS "R-u"                    {*+u=*}
QS "R-uu"                   {*+uu=*}
QS "R-o"                    {*+o=*}
QS "R-oo"                   {*+oo=*}
QS "R-one"                  {*+one=*}
QS "R-oneone"               {*+oneone=*}
QS "R-a"                    {*+a=*}
QS "R-aa"                   {*+aa=*}
QS "R-p"                    {*+p=*}
QS "R-b"                    {*+b=*}
QS "R-t"                    {*+t=*}
QS "R-d"                    {*+d=*}
QS "R-c"                    {*+c=*}
QS "R-gj"                   {*+gj=*}
QS "R-k"                    {*+k=*}
QS "R-g"                    {*+g=*}
QS "R-f"                    {*+f=*}
QS "R-v"                    {*+v=*}
QS "R-s"                    {*+s=*}
QS "R-z"                    {*+z=*}
QS "R-S"                    {*+S=*}
QS "R-Z"                    {*+Z=*}
QS "R-h"                    {*+h=*}
QS "R-G"                    {*+G=*}
QS "R-tS"                   {*+tS=*}
QS "R-dZ"                   {*+dZ=*}
QS "R-m"                    {*+m=*}
QS "R-n"                    {*+n=*}
QS "R-l"                    {*+l=*}
QS "R-five"                 {*+five=*}
QS "R-r"                    {*+r=*}
QS "R-j"                    {*+j=*}
QS "R-w"                    {*+w=*}
QS "R-N"                    {*+N=*}
QS "R-pau"                  {*+pau=*}
QS "R-brth"                 {*+brth=*}
QS "R-wb"                  {*+wb=*}

QS "RR-Vowel"              {*=i@*,*=e@*,*=y@*,*=yy@*,*=two@*,*=twotwo@*,*=u@*,*=o@*,*=one@*,
                            *=a@*,*=ii@*,*=ee@*,*=uu@*,*=oo@*,*=oneone@*,*=aa@*}
QS "RR-Consonant"          {*=p@*,*=b@*,*=t@*,*=d@*,*=c@*,*=gj@*,*=k@*,*=g@*,*=f@*,*=v@*,
                            *=s@*,*=z@*,*=S@*,*=Z@*,*=h@*,*=G@*,*=tS@*,*=dZ@*,*=m@*,*=n@*,
                            *=l@*,*=five@*,*=r@*,*=j@*,*=w@*,*=N@*}
QS "RR-Stop"               {*=p@*,*=b@*,*=t@*,*=d@*,*=c@*,*=gj@*,*=k@*,*=g@*,*=tS@*,*=dZ@*}
QS "RR-Nasal"              {*=m@*,*=n@*,*=N@*}
QS "RR-Fricative"          {*=f@*,*=v@*,*=s@*,*=z@*,*=S@*,*=Z@*,*=h@*,*=G@*}
QS "RR-Liquid"             {*=l@*,*=five@*,*=r@*}
QS "RR-Front"              {*=i@*,*=ii@*,*=e@*,*=ee@*,*=y@*,*=yy@*,*=two@*,*=twotwo@*,*=p@*,
                            *=b@*,*=m@*,*=f@*,*=v@*,*=m@*,*=w@*}
QS "RR-Central"            {*=d@*,*=dZ@*,*=l@*,*=five@*,*=n@*,*=r@*,*=s@*,*=S@*,*=t@*,*=z@*,
                            *=Z@*}
QS "RR-Back"               {*=u@*,*=uu@*,*=o@*,*=oo@*,*=one@*,*=oneone@*,*=a@*,*=aa@*,*=c@*,
                            *=gj@*,*=g@*,*=G@*,*=h@*,*=j@*,*=k@*}
QS "RR-Front_Vowel"        {*=i@*,*=ii@*,*=e@*,*=ee@*,*=y@*,*=yy@*,*=two@*,*=twotwo@*}
QS "RR-Back_Vowel"         {*=u@*,*=uu@*,*=o@*,*=oo@*,*=one@*,*=oneone@*,*=a@*,*=aa@*}
QS "RR-Long_Vowel"         {*=ii@*,*=ee@*,*=uu@*,*=oo@*,*=yy@*,*=oneone@*,*=aa@*,*=twotwo@*}
QS "RR-Short_Vowel"        {*=i@*,*=e@*,*=y@*,*=two@*,*=u@*,*=o@*,*=one@*,*=a@*}
QS "RR-Front_Start_Vowel"  {*=i@*,*=ii@*,*=e@*,*=ee@*,*=y@*,*=yy@*,*=two@*,*=twotwo@*}
QS "RR-High_Vowel"         {*=i@*,*=ii@*,*=y@*,*=yy@*,*=u@*,*=uu@*,*=one@*,*=oneone@*}
QS "RR-Medium_Vowel"       {*=e@*,*=ee@*,*=two@*,*=twotwo@*,*=o@*,*=oo@*}
QS "RR-Low_Vowel"          {*=a@*,*=aa@*}
QS "RR-Rounded_Vowel"      {*=w@*}
QS "RR-Unrounded_Vowel"    {*=i@*,*=ii@*,*=e@*,*=ee@*,*=y@*,*=yy@*,*=two@*,*=twotwo@*,*=u@*,
                            *=uu@*,*=o@*,*=oo@*,*=one@*,*=oneone@*,*=a@*,*=aa@*}
QS "RR-IVowel"             {*=i@*,*=ii@*,*=y@*,*=yy@*}
QS "RR-EVowel"             {*=e@*,*=ee@*}
QS "RR-AVowel"             {*=a@*,*=aa@*}
QS "RR-OVowel"             {*=o@*,*=oo@*,*=two@*,*=twotwo@*}
QS "RR-UVowel"             {*=u@*,*=uu@*,*=one@*,*=oneone@*}
QS "RR-Unvoiced_Consonant" {*=p@*,*=t@*,*=c@*,*=k@*,*=f@*,*=s@*,*=S@*,*=h@*,*=tS@*}
QS "RR-Voiced_Consonant"   {*=b@*,*=d@*,*=gj@*,*=g@*,*=v@*,*=z@*,*=Z@*,*=G@*,*=dZ@*,*=m@*,
                            *=n@*,*=l@*,*=five@*,*=r@*,*=j@*,*=w@*,*=N@*}
QS "RR-Front_Consonant"    {*=p@*,*=b@*,*=m@*,*=f@*,*=v@*,*=m@*,*=w@*}
QS "RR-Central_Consonant"   {*=d@*,*=dZ@*,*=l@*,*=five@*,*=n@*,*=r@*,*=s@*,*=S@*,*=t@*,
                             *=z@*,*=Z@*}
QS "RR-Back_Consonant"      {*=c@*,*=gj@*,*=g@*,*=G@*,*=h@*,*=j@*,*=k@*}
QS "RR-Coronal_Consonant"   {*=t@*,*=d@*,*=s@*,*=z@*,*=S@*,*=Z@*,*=tS@*,*=dZ@*,*=n@*,*=r@*,
                             *=N@*,*=l@*,*=five@*}
QS "RR-Non_Coronal"         {*=p@*,*=b@*,*=f@*,*=c@*,*=gj@*,*=k@*,*=g@*,*=f@*,*=v@*,*=h@*,
                             *=G@*,*=m@*,*=j@*,*=w@*}
QS "RR-Anterior_Consonant"  {*=b@*,*=d@*,*=f@*,*=l@*,*=five@*,*=m@*,*=n@*,*=N@*,*=p@*,
                             *=s@*,*=t@*,*=v@*,*=w@*,*=z@*}
QS "RR-Non_Anterior"        {*=tS@*,*=g@*,*=h@*,*=Z@*,*=c@*,*=k@*,*=r@*,*=S@*,*=y@*,*=yy@*,
                            *=dZ@*}
QS "RR-Continuent"          {*=f@*,*=v@*,*=s@*,*=z@*,*=S@*,*=Z@*,*=h@*,*=G@*,*=l@*,
                             *=five@*,*=r@*,*=j@*,*=w@*}
QS "RR-No_Continuent"       {*=p@*,*=b@*,*=t@*,*=d@*,*=c@*,*=gj@*,*=k@*,*=g@*,*=tS@*,
                             *=dZ@*,*=m@*,*=n@*, *=N@*}
QS "RR-Glide"               {*=h@*,*=l@*,*=five@*,*=r@*,*=y@*,*=yy@*,*=w@*}
QS "RR-Voiced_Stop"         {*=b@*,*=d@*,*=g@*,*=gj@*,*=dZ@*}
QS "RR-Unvoiced_Stop"       {*=p@*,*=t@*,*=k@*,*=c@*,*=tS@*}
QS "RR-Front_Stop"          {*=b@*,*=p@*}
QS "RR-Central_Stop"        {*=d@*,*=t@*,*=tS@*,*=dZ@*}
QS "RR-Back_Stop"           {*=g@*,*=k@*,*=c@*}
QS "RR-Voiced_Fricative"    {*=v@*,*=z@*,*=Z@*,*=G@*}
QS "RR-Unvoiced_Fricative"  {*=f@*,*=s@*,*=S@*,*=h@*}
QS "RR-Front_Fricative"     {*=f@*,*=v@*}
QS "RR-Central_Fricative"   {*=s@*,*=z@*}
QS "RR-Back_Fricative"      {*=S@*,*=Z@*}
QS "RR-Affricate_Consonant" {*=tS@*,*=dZ@*}
QS "RR-silences"            {*=pau@*,*=h#@*,*=brth@*,*=wb@*}
QS "RR-i"                   {*=i@*}
QS "RR-ii"                  {*=ii@*}
QS "RR-e"                   {*=e@*}
QS "RR-ee"                  {*=ee@*}
QS "RR-y"                   {*=y@*}
QS "RR-yy"                  {*=yy@*}
QS "RR-two"                 {*=two@*}
QS "RR-twotwo"              {*=twotwo@*}
QS "RR-u"                   {*=u@*}
QS "RR-uu"                  {*=uu@*}
QS "RR-o"                   {*=o@*}
QS "RR-oo"                  {*=oo@*}
QS "RR-one"                 {*=one@*}
QS "RR-oneone"              {*=oneone@*}
QS "RR-a"                   {*=a@*}
QS "RR-aa"                  {*=aa@*}
QS "RR-p"                   {*=p@*}
QS "RR-b"                   {*=b@*}
QS "RR-t"                   {*=t@*}
QS "RR-d"                   {*=d@*}
QS "RR-c"                   {*=c@*}
QS "RR-gj"                  {*=gj@*}
QS "RR-k"                   {*=k@*}
QS "RR-g"                   {*=g@*}
QS "RR-f"                   {*=f@*}
QS "RR-v"                   {*=v@*}
QS "RR-s"                   {*=s@*}
QS "RR-z"                   {*=z@*}
QS "RR-S"                   {*=S@*}
QS "RR-Z"                   {*=Z@*}
QS "RR-h"                   {*=h@*}
QS "RR-G"                   {*=G@*}
QS "RR-tS"                  {*=tS@*}
QS "RR-dZ"                  {*=dZ@*}
QS "RR-m"                   {*=m@*}
QS "RR-n"                   {*=n@*}
QS "RR-l"                   {*=l@*}
QS "RR-five"                {*=five@*}
QS "RR-r"                   {*=r@*}
QS "RR-j"                   {*=j@*}
QS "RR-w"                   {*=w@*}
QS "RR-N"                   {*=N@*}
QS "RR-pau"                 {*=pau@*}
QS "RR-brth"                {*=brth@*}
QS "RR-wb"                  {*=wb@*}
QS "C-Syl_Vowel==x"        {*|x/C:*}
QS "C-Syl_Vowel==no"       {*|novowel/C:*}
QS "C-Syl_Vowel"           {*|i/C:*,*|e/C:*,*|y/C:*,*|yy/C:*,*|two/C:*,*|twotwo/C:*,
                            *|u/C:*,*|o/C:*,*|one/C:*,*|a/C:*,*|ii/C:*,*|ee/C:*,*|uu/C:*,
                            *|oo/C:*,*|oneone/C:*,*|aa/C:*}
QS "C-Syl_Front_Vowel"     {*|i/C:*,*|ii/C:*,*|e/C:*,*|ee/C:*,*|y/C:*,*|yy/C:*,*|two/C:*,
                            *|twotwo/C:*}
QS "C-Syl_Back_Vowel"      {*|u/C:*,*|uu/C:*,*|o/C:*,*|oo/C:*,*|one/C:*,*|oneone/C:*,
                            *|a/C:*,*|aa/C:*}
QS "C-Syl_Long_Vowel"      {*|ii/C:*,*|ee/C:*,*|uu/C:*,*|oo/C:*,*|oneone/C:*,*|aa/C:*,
                            *|twotwo/C:*,*|yy/C:*}
QS "C-Syl_Short_Vowel"     {*|i/C:*,*|e/C:*,*|y/C:*,*|two/C:*,*|u/C:*,*|o/C:*,*|one/C:*,
                            *|a/C:*}
QS "C-Syl_Front_Start"     {*|i/C:*,*|ii/C:*,*|e/C:*,*|ee/C:*,*|y/C:*,*|yy/C:*,*|two/C:*,
                            *|twotwo/C:*}
QS "C-Syl_High_Vowel"      {*|i/C:*,*|ii/C:*,*|y/C:*,*|yy/C:*,*|u/C:*,*|uu/C:*,*|one/C:*,
                            *|oneone/C:*}
QS "C-Syl_Medium_Vowel"    {*|e/C:*,*|ee/C:*,*|two/C:*,*|twotwo/C:*,*|o/C:*,*|oo/C:*}
QS "C-Syl_Low_Vowel"       {*|a/C:*,*|aa/C:*}
QS "C-Syl_Rounded_Vowel"   {*|w/C:*}
QS "C-Syl_Unrounded_Vowel" {*|i/C:*,*|ii/C:*,*|e/C:*,*|ee/C:*,*|y/C:*,*|yy/C:*,*|two/C:*,
                            *|twotwo/C:*,*|u/C:*,*|uu/C:*,*|o/C:*,*|oo/C:*,*|one/C:*,
                            *|oneone/C:*,*|a/C:*,*|aa/C:*}
QS "C-Syl_IVowel"          {*|i/C:*,*|ii/C:*,*|y/C:*,*|yy/C:*}
QS "C-Syl_EVowel"          {*|e/C:*,*|ee/C:*}
QS "C-Syl_AVowel"          {*|a/C:*,*|aa/C:*}
QS "C-Syl_OVowel"          {*|o/C:*,*|oo/C:*,*|two/C:*,*|twotwo/C:*}
QS "C-Syl_UVowel"          {*|u/C:*,*|uu/C:*,*|one/C:*,*|oneone/C:*}
QS "C-Syl_i"               {*|i/C:*}
QS "C-Syl_ii"              {*|ii/C:*}
QS "C-Syl_e"               {*|e/C:*}
QS "C-Syl_ee"              {*|ee/C:*}
QS "C-Syl_y"               {*|y/C:*}
QS "C-Syl_yy"              {*|yy/C:*}
QS "C-Syl_two"             {*|two/C:*}
QS "C-Syl_twotwo"          {*|twotwo/C:*}
QS "C-Syl_u"               {*|u/C:*}
QS "C-Syl_uu"               {*|uu/C:*}
QS "C-Syl_o"                {*|o/C:*}
QS "C-Syl_oo"               {*|oo/C:*}
QS "C-Syl_one"              {*|one/C:*}
QS "C-Syl_oneone"           {*|oneone/C:*}
QS "C-Syl_a"                {*|a/C:*}
QS "C-Syl_aa"               {*|aa/C:*}

CQS "Seg_Fw"                {@(\d+)_}
CQS "Seg_Bw"                {_(\d+)/A:}
CQS "R-Syl_Num-Segs"        {+(\d+)/D:}
QS "L-Word_GPOS==0"         {/D:0_}
QS "L-Word_GPOS==aux"       {*/D:aux_*}
QS "L-Word_GPOS==cc"        {*/D:cc_*}
QS "L-Word_GPOS==content"   {*/D:content_*}
QS "L-Word_GPOS==det"       {*/D:det_*}
QS "L-Word_GPOS==in"        {*/D:in_*}
QS "L-Word_GPOS==md"        {*/D:md_*}
QS "L-Word_GPOS==pps"       {*/D:pps_*}
QS "L-Word_GPOS==punc"      {*/D:punc_*}
QS "L-Word_GPOS==to"        {*/D:to_*}
QS "L-Word_GPOS==wp"        {*/D:wp_*}
CQS "L-Word_Num-Syls"       {_(\d+)/E:}
QS "C-Word_GPOS==x"         {/E:x+}
QS "C-Word_GPOS==aux"       {*/E:aux+*}
QS "C-Word_GPOS==cc"        {*/E:cc+*}
QS "C-Word_GPOS==content"   {*/E:content+*}
QS "C-Word_GPOS==det"       {*/E:det+*}
QS "C-Word_GPOS==in"        {*/E:in+*}
QS "C-Word_GPOS==md"        {*/E:md+*}
QS "C-Word_GPOS==pps"       {*/E:pps+*}
QS "C-Word_GPOS==punc"      {*/E:punc+*}
QS "C-Word_GPOS==to"        {*/E:to+*}
QS "C-Word_GPOS==wp"        {*/E:wp+*}
CQS "C-Word_Num-Syls"       {+(\d+)@}
CQS "Pos_C-Word_in_C-Phrase(Fw)"             {@(\d+)+}
CQS "Pos_C-Word_in_C-Phrase(Bw)"             {+(\d+)&}
CQS "Num-ContWord_before_C-Word_in_C-Phrase" {&(\d+)+}
CQS "Num-ContWord_after_C-Word_in_C-Phrase"  {+(\d+)#}
CQS "Num-Words_from_prev-ContWord"           {#(\d+)+}
CQS "Num-Words_from_next-ContWord"           {+(\d+)/F:}
QS "R-Word_GPOS==aux"       {*/F:aux_*}
QS "R-Word_GPOS==cc"        {*/F:cc_*}
QS "R-Word_GPOS==content"   {*/F:content_*}
QS "R-Word_GPOS==det"       {*/F:det_*}
QS "R-Word_GPOS==in"        {*/F:in_*}
QS "R-Word_GPOS==md"        {*/F:md_*}
QS "R-Word_GPOS==pps"       {*/F:pps_*}
QS "R-Word_GPOS==punc"      {*/F:punc_*}
QS "R-Word_GPOS==to"        {*/F:to_*}
QS "R-Word_GPOS==wp"        {*/F:wp_*}
QS "R-Word_GPOS==0"         {/F:0_}
CQS "R-Word_Num-Syls"       {_(\d+)/G:}
CQS "L-Phrase_Num-Syls"     {/G:(\d+)_}
CQS "L-Phrase_Num-Words"    {_(\d+)/H:}
CQS "C-Phrase_Num-Syls"     {/H:(\d+)=}
CQS "C-Phrase_Num-Words"    {=(\d+)@}
CQS "Pos_C-Phrase_in_Utterance(Fw)" {@(\d+)=}
CQS "Pos_C-Phrase_in_Utterance(Bw)" {=(\d+)&}
CQS "R-Phrase_Num-Syls"     {/I:(\d+)=}
CQS "R-Phrase_Num-Words"    {=(\d+)/J:}
CQS "Num-Syls_in_Utterance" {/J:(\d+)+}
CQS "Num-Words_in_Utterance"        {+(\d+)-}
CQS "Num-Phrases_in_Utterance"      {-(\d+)}
```