Recovering Discrete Prosody Inputs via Invert-Classify

Nicholas Sanders1, Korin Richmond2

1 CSTR, University of Edinburgh, Edinburgh, United Kingdom
2 CSE, University of Illinois at Urbana Champaign, Urbana, Illinois, USA

Abstract

Modeling prosody in Text-to-Speech (TTS) is challenging due to ambiguous orthography and the high cost of annotating prosodic events. This study focuses on the modeling of contrastive focus, the emphasis of a word to contrast it to presuppositions held by an interlocutor. Modeling of contrastive focus can be done in TTS by using binary, symbolic inputs at the word level in a supervised setting. To address the absence of annotated data, we propose the Invert-Classify method, which leverages a frozen TTS model and unlabeled parallel text-speech data to recover missing contrastive focus inputs. Our approach achieves a binary F-score of 0.71 for contrastive focus annotation recovery, utilizing only 10% of annotated training data. These findings establish fundamental insights and techniques that can be extended and refined for other prosody modeling methods in TTS.

Index Terms: text-to-speech synthesis, prosody modeling

1. Introduction

Text-to-Speech (TTS) data typically consists of parallel text and speech waveform data. However, most text is plain and does not specify all properties of speech, such as prosodic properties (e.g., rhythm, intonation, loudness, etc.). To address this ambiguity and enhance the synthesis of natural and expressive speech, TTS systems commonly employ one of two approaches: incorporating a reference utterance for style transfer [1], and incorporating symbolic inputs alongside text to further specify prosody [2]. This work focuses on the latter method for contrastive focus in English (i.e., the emphasis of words to contrast presuppositions of an interlocutor). Contrastive focus in English is realized by modifying acoustic correlates such as: F0, duration, and energy, in order to draw attention to a word that corrects contextual information.

Contrastive focus for TTS can be modeled at the word level, with binary annotations that denote whether a word should have contrastive focus or not. However, these annotations can be costly to derive from human listeners. Therefore, we aim to automatically derive contrastive focus labels for parallel text-speech data, using our proposed method called Invert-Classify, further described in Section 3. We demonstrate the effectiveness of Invert-Classify in conjunction with FastPitch models [3] trained on varying amounts of training data. These findings showcase the potential of enhancing the efficiency of TTS models that utilize symbolic inputs for prosody specification, thereby reducing the required annotation burden.

2. Data and Modeling

2.1. Data

The data used in this study is from the Naver Prosody-Control data set [4], which features 26.7 hours of 36600 individual speech utterances with corresponding text, from an American English female speaker. Different versions or minimal pair groups of utterances were recorded with neutral (i.e., declarative), questioning, contrastive focus on the subject, contrastive focus on the object, and contrastive focus on the verb. However, the question utterances in the data set were excluded in this study. During recording, utterances were presented to the speaker with a context prompt, which assisted in specifying prosody further. A sample of a minimal pair group in writing can be seen in Table 1. For this work, we derive the train, validation, and test sets via an 80-10-10 split.

2.2. Model Description

The TTS model used in this work is known as FastPitch, a transformer-based sequence to sequence model, which in our implementation receives arpabet phone symbols (attained during forced alignment) as inputs and predicts corresponding frames of mel-spectrogram. During training, FastPitch receives pitch and duration targets obtained during preprocessing for each phone symbol in the training data. The modeling of pitch and duration targets is incorporated in the final fusion layer, along with L2 loss of predicted and ground truth mel-spectrograms.

2.3. Contrastive Focus Modeling

Similar to previous works, FastPitch receives additional text input in the form of binary symbols, indicating whether a phone occurs within a word that has contrastive focus or not [4]. This means that each phone has a corresponding annotation that indicates focus with the arbitrarily assigned values of 2 for no contrastive focus, or 1 for contrastive focus, and 0 reserved for padding. The contrastive focus annotations are passed through a separate embedding layer before being summed with the em-

Table 1: Sample from the Naver Prosody-Control Dataset

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral</td>
<td>Abby went home.</td>
</tr>
<tr>
<td>Question</td>
<td>Abby went home?</td>
</tr>
<tr>
<td>Cara went home?</td>
<td>ABBY went home.</td>
</tr>
<tr>
<td>Abby dax home?</td>
<td>Abby WENT home.</td>
</tr>
<tr>
<td>Abby went going?</td>
<td>Abby went HOME.</td>
</tr>
</tbody>
</table>
beddings of the phone inputs and the positional embeddings.

2.4. Training Regime
FastPitch is trained by optimizing a combination of L2 losses between the predicted and ground truth mel spectrogram, durations, and pitch values. However, the duration and pitch losses are scaled by a weight, which acts as a hyperparameter, and then are subsequently added to the mel spectrogram loss. During training, batch size of 30 is used, LAMB is used as an optimization algorithm, 0.2 is set as the weight multiplied with the loss for duration and pitch, and an overall learning rate of 0.1 is used. All Fastpitch models were trained for 1000 epochs.

3. Inversion
After training FastPitch, minimal pair groups that do not appear in the training set are selected for inversion. Inversion is the process of performing back propagation on a frozen model and updating inputs to minimize some loss. Originally, inversion was intended for visualizing the generalization of classification models [5]. However, recently inversion has seen some success in tasks such as text to image, where new style embeddings can be generated via inversion to represent new style tokens [6]. In this case, the inputs are the contrastive focus embeddings and they are optimized via Adam with a learning rate of 0.75 to minimize the same loss used during the training of FastPitch.

3.1. Cosine classification, Initialization, and Metrics
The initialization of the learnable embeddings via inversion is obtained by taking the mean of the two embeddings within the embedding table for contrastive focus. While inverting the embeddings, the previously learned embeddings from the embedding table serve as two basis vectors, then a cosine similarity along with an argmax is calculated to hard assign all inputs to one of the two discrete values. After cosine classification, accuracy metrics can be obtained by comparing to the ground truth contrastive focus annotations.

3.2. Inductive Biases
Different conditional statements have been implemented to constrain the inversion and quicken convergence. Not allowing silence to be updated and tying all embeddings that correspond to the same word to one single embedding achieved better results for accuracy and convergence speed. Therefore, we implement these inductive biases for our experiments.

4. Experiment and Results
To select when to stop inversion, we select the epoch with the lowest fusion L2 loss. We find that this achieves better accuracy results on average.

The effect of training set size on classification accuracy is explored by removing minimal pair groupings to form subsets which are 1%, 5%, 10%, 20%, and 100% of the original training set size. As seen in Table 2, binary fscore (with contrastive focus being the positive label) begins to drop steeply below 10% of the training set size. Recall is comparatively higher than precision in all cases, meaning that inaccuracies are likely due to over predicting of contrastive focus. The high false positive rate may be affected by a lower occurrence of contrastive focus and noise within ground truth annotations, however that is unlikely to explain all of the false positives.

<table>
<thead>
<tr>
<th>Training Set Size</th>
<th>Precision</th>
<th>Recall</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>0.42</td>
<td>0.73</td>
<td>0.54</td>
</tr>
<tr>
<td>5%</td>
<td>0.51</td>
<td>0.919</td>
<td>0.66</td>
</tr>
<tr>
<td>10%</td>
<td>0.58</td>
<td>0.93</td>
<td>0.71</td>
</tr>
<tr>
<td>20%</td>
<td>0.56</td>
<td>0.94</td>
<td>0.70</td>
</tr>
<tr>
<td>100%</td>
<td>0.61</td>
<td>0.92</td>
<td>0.74</td>
</tr>
</tbody>
</table>

5. Comments and Future Work
Further error analysis is yet to be explored, although many false positives have been observed to be adjacent words to words that contain contrastive focus (e.g., ‘the’ preceding a noun with contrastive focus). We are currently investigating what the impact of continuing to train the model on the data that was labeled via Invert-Classify. While comprehensive large-scale listening tests are pending, preliminary observations suggest that training on additional data labeled via Invert-Classify enhances overall synthesis quality. It is worth noting that synthesizing with inverted embeddings as inputs without applying classification yields unsatisfactory results.

One limitation is that the data set used in the experiments is constrained to short utterances that follow a particular subject-object-verb order, and performance may be different on less ideal data that has a lower frequency of contrastive focus. Finally, we plan on exploring further optimization constraints, inductive biases, and alternative loss functions that may assist the inversion process.

6. Acknowledgments
This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1), the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences, and Huawei.

7. References