NEURAL SPEECH-TO-SPEECH SYNTHESIS

Jared Samet - UNI: jss2272

jss2272@columbia.edu

ABSTRACT

Prosody features that transfer across speakers can be extracted using unsupervised learning. By adding the cluster labels as annotation to text, a text-to-speech synthesis system can learn to incorporate the prosody into its output. The resulting system can produce audio that is different even when the text is the same and the resulting audio can mimic the prosody from the original speaker. The abstract should be about 175 words total.

Index Terms— Prosody, unsupervised learning, speech synthesis, seq2seq

1. INTRODUCTION

I used Kaldi to create an alignment for the Tedlium data using the final triphone model it created. For each vowel phoneme, I used Kaldi's pitch extractor and the first MFCC component (energy) to create two series of numbers. Kaldi's pitch extractor is already normalized but I used [StandardScaler] to normalize the energy component across the utterance. I fit a second-degree (?) Legendre polynomial to the pitch and power to create six features for each vowel. The duration gave me the seventh feature. I then ran K-means clustering on these to create eight different vowel clusters that were common across the entire range of speakers in the (subsampled) Tedlium data.

I then used the same triphone model to generate an alignment for the LJSpeech dataset, extracted the same pitch, power, and duration features for LJ, and used the previously computed vowel clusters to assign a cluster label to each vowel in the LJSpeech dataset. I (slightly) modified the Tacotron implementation to accept a sequence of tokens instead of a sequence of characters. Instead of text characters, my input tokens consisted of (Kaldi's) phonemes and the vowel cluster labels. Having suitably modified Tacotron, I then trained Tacotron on the [phoneme + cluster label, audio] pairs.

Finally, to see if it worked, I recorded myself saying a sentence in multiple ways, ran each .wav through the same align + label steps, and fed the resulting [phoneme+label, text] pairs to Tacotron. She said the same thing different ways.

2. RELATED WORK

This project involved two main components: first, extracting prosody features from a set of input audio files; and second, training a text-to-speech synthesis model on a dataset that had been labeled using the extracted prosody features.

Selkirk [1] discusses sentence prosody and pitch accent in the context of English. Although English is generally not thought of as a tonal language, Selkirk writes that "[i]n English a pitch accent associates to a stress-prominent syllable in a word (typicall the main word stress)." Fujisaki [2] models the F_0 contour over the duration of an utterance as the sum of a set of impulse response and step response functions, parameterized with a finite number of scalar values. Wang et al. [3] use the pitch and amplitude contours to improve tone recognition in Mandarin by identifying "maxima, minima, and inflection points of particular acoustic events." Wong and Siu [4] use robust regression and orthogonal polynomials to create features for a decision tree classifier in order to recognize tones in Chinese lan guages. Finally, Lin [5] and Mary [6] use a small number of Legendre polynomial coefficients to represent the pitch contour as a finite-dimensional feature vector, which is the approach used in this project.

Speech synthesis or text-to-speech is a well-studied problem that has been actively researched since the 1950s. While there has been remarkable progress in the field in recent years, the quality of computer-generated speech has not yet reached human levels. Current commercial systems described in Khan et al. [7] and Taylor [8] generally use concatenative speech synthesis to produce their output. However, the alternative approach of parametric synthesis using neural networks is rapidly gaining popularity, with several papers since 2016 demonstrating impressive results in the quality of the output. The first of this generation was Google's WaveNet (Oord et al. [9]), followed in quick succession by Deep Voice and Deep Voice 2 from Baidu (Arik et al. [10], [11]), Char2Wav from MILA (Sotelo et al. [12]), and Tacotron from Google (Wang et al. [13]). Each of these systems has taken a different approach to the network architecture to address different aspects of the speech synthesis pipeline.

3. PROSODY FEATURES

I used pitch (from Kaldi) and power. I used first three Legendre coefficients to extract a finite set of features for a phoneme of arbitrary length by using [-1, 1] as the domain regardless of the actual length of the phoneme. I added a few frames at the beginning and end in case Kaldi got the alignment wrong. Show what the Legendre coefficients look like for some different curves. Show what the actual pitch and power curves look like for some brief, manually labeled utterances. Describe the Kaldi pitch extractor. Definitely include the Kaldi paper as a reference. Explain how the Legendre coefficients work. Explain why these were a sensible way of capturing prosody. Talk about pitch envelopes and tonal languages.

4. CLUSTERING VOWELS

Unsupervised learning FTW. Discuss why we should expect there to be clusters even in an atonal language like English. Talk about stressed vs unstressed as initial motivation but how there are probably more things like this. Talk about how stress is mostly a pitch change.

I ran K-means clustering on my seven features and created eight vowel clusters. I'm pretty sure I ran StandardScaler on the features first so K-means didn't get confused, double check this. This was intended to be enough to capture the major variation across how different vowels can be pronounced but not so many as to result in too-few training examples. I originally did this for each vowel separately, and only for a single speaker, but then I decided that was stupid so I did it across all speakers and across eight vowels. So there are only eight clusters. Here I need to demonstrate that the clusters are in fact "semantically" different in some way. Maybe include some metric of these or run TSNE on the coefficients.

I could probably have also just used the actual Legendre coefficients themselves but this would have required tinkering with the Tacotron internals more to accept continuous-valued features as part of the sequence instead of just a one-hot encoded value. This is something that could go in a future work section.

5. TACOTRON

Describe the Tacotron architecture and explain why it was easy to add the cluster labels.

5.1. Subheadings

Just for reminder.

5.1.1. Sub-subheadings

Just for reminder.

6. PIPELINE

Explain the full speech-to-speech pipeline:

First is the training process, whose outputs are a vowel cluster model and a trained Tacotron

- Use Kaldi to create alignments and pitch/power features from Tedlium
- Use my code [part 1] to compute 7 features for each vowel
- Use my code [part 2] to create a vowel cluster model
- Use Kaldi to create alignments and pitch/power features from LJSpeech
- Use my code [part 1] to compute 7 features for each vowel
- Use my code [part 3] to assign cluster labels to each vowel
- Train Tacotron on annotated LJSpeech

Second is the speech-to-speech pipeline, whose input is the vowel cluster model and the trained Tacotron from part 1, plus the WAV from a new speaker

- Use Kaldi to create alignments and pitch/power features for the new WAV
- Use my code [part 1] to compute 7 features for each vowel
- Use my code [part 3] to assign cluster labels to each vowel
- Run the phones + cluster labels through the trained Tacotron to produce the output WAV

7. RESULTS

Find some way to quantify that it actually did something beyond "she never stole my money".

Try and quantify that the different clusters are actually different in some way. This is probably the most important section. Quantify if they are different from male to female speakers in any way.

Try and quantify that the speech result is better for my Tacotron than for without annotations. Say why this could be useful even if no one wants to do speech to speech.

Try and quantify that the output is actually preserving stuff from the original speech dataset.

8. DISCUSSION

9. LIMITATIONS

10. FUTURE WORK

11. ILLUSTRATIONS, GRAPHS, AND PHOTOGRAPHS

Illustrations must appear within the designated margins. They may span the two columns. If possible, position illustrations at the top of columns, rather than in the middle or at the bottom. Caption and number every illustration. All halftone illustrations must be clear black and white prints. Colors may be used, but they should be selected so as to be readable when printed on a black-only printer.

Since there are many ways, often incompatible, of including images (e.g., with experimental results) in a LaTeX document, below is an example of how to do this.

12. FOOTNOTES

Use footnotes sparingly (or not at all!) and place them at the bottom of the column on the page on which they are referenced. Use Times 9-point type, single-spaced. To help your readers, avoid using footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence).

(a) Result 1

(b) Results 3 (c) Result 4

Fig. 1. Example of placing a figure with experimental results.

13. COPYRIGHT FORMS

You must submit your fully completed, signed IEEE electronic copyright release form when you submit your paper. We **must** have this form before your paper can be published in the proceedings.

14. RELATION TO PRIOR WORK

The text of the paper should contain discussions on how the paper's contributions are related to prior work in the field. It is important to put new work in context, to give credit to foundational work, and to provide details associated with the previous work that have appeared in the literature. This discussion may be a separate, numbered section or it may appear elsewhere in the body of the manuscript, but it must be present.

You should differentiate what is new and how your work expands on or takes a different path from the prior studies. An example might read something to the effect: "The work presented here has focused on the formulation of the ABC algorithm, which takes advantage of non-uniform time-frequency domain analysis of data. The work by Smith and Cohen considers only fixed time-domain analysis and the work by Jones et al takes a different approach based on fixed frequency partitioning. While the present study is related to recent approaches in time-frequency analysis [3-5], it capitalizes on a new feature space, which was not considered in these earlier studies."

15. REFERENCES

List and number all bibliographical references at the end of the paper. The references can be numbered in alphabetic order or in order of appearance in the document. When referring to them in the text, type the corresponding reference number in square brackets as shown at the end of this sentence [2]. An additional final page (the fifth page, in most cases) is allowed, but must contain only references to the prior literature.

16. REFERENCES

[1] Elisabeth Selkirk, "Sentence prosody: Intonation, stress, and phrasing," *The handbook of phonological theory*, vol. 1, pp. 550–569, 1995.

[2] Hiroya Fujisaki, "Information, prosody, and modeling-with emphasis on tonal features of speech," in *Speech Prosody 2004, International Conference*, 2004.

[3] Siwei Wang and Gina-Anne Levow, "Mandarin chinese tone nucleus detection with landmarks," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[4] WONG Pui-Fung and SIU Man-Hung, "Decision tree based tone modeling for chinese speech recognition," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 1, pp. I–905.

[5] Chi-Yueh Lin and Hsiao-Chuan Wang, "Language identification using pitch contour information," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. IEEE, 2005, vol. 1, pp. I–601.

[6] Leena Mary, *Extraction and representation of prosody for speaker, speech and language recognition*, Springer Science & Business Media, 2011.

[7] Rubeena A Khan and JS Chitode, "Concatenative speech synthesis: A review," *International Journal of Computer Applications*, vol. 136, no. 3, pp. 6, 2016.

[8] Paul Taylor, *Text-to-speech synthesis*, Cambridge university press, 2009.

[9] Aaron 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," *arXiv preprint arXiv:1609.03499*, 2016.

[10] Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, et al., "Deep voice: Real-time neural text-to-speech," *arXiv preprint arXiv:1702.07825*, 2017.

[11] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.

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

[13] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: A fully end-to-end text-to-speech synthesis model," *arXiv preprint arXiv:1703.10135*, 2017.