Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Master thesis

Phonetic Representations of Speech for Human Pronunciation Feedback and Automatic Accent Transfer

Isaac Riley

Studiengang: M.Sc. Computational Linguistics

Prüfer*innen: Prof. Dr. Wolfgang Wokurek
Prof. Dr. Antje Schweitzer

Betreuer: Prof. Dr. Wolfgang Wokurek

Beginn der Arbeit: 01.04.2021

Ende der Arbeit: 01.10.2021

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht. Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein.[1]

(Isaac Riley)

[1]Non-binding translation for convenience: This thesis is the result of my own independent work, and any material from work of others which is used either verbatim or indirectly in the text is credited to the author including details about the exact source in the text. This work has not been part of any other previous examination, neither completely nor in parts. It has neither completely nor partially been published before. The submitted electronic version is identical to this print version.

Acknowledgments
XXX

Inhaltsverzeichnis

1 Introduction

Accent

2 Background

3 Related Work

4 Resources

5 Methods

5.1 Phonetic Representations

To learn representations of input speech samples, I use a neural network, or rather a family of similar neural networks all sharing the same sequence-to-sequence architecture. This architecture consists of an encoder, a decoder, and an attention mechanism, each of which will be discussed in greater detail below.

5.1.1 Phone Recognition Model

Encoder The audio files in .wav format are pre-processed into 80-dimensional filter banks to be input into the encoder. Additionally, in some experiments between 2 and 4 frames are "stacked". Thus, for an input sample with N frames, stacking would result in and $N/2$ frames each with dimensionality of 160; stacking 3 frames results in $N/3$ frames each with dimensionality 240, and so on.

The encoder is a

Decoder The initial decoder state, like the initial encoder state, contains learnable parameters; each successive state is derived from the previous decoder state and from the encoder output.

At time step t, the decoder's task is to predict the next output, $\hat{y}_t$. To do this it uses the embedding of the last prediction, $m_t = W^E(\hat{y}_{t-1})$.

$$\mathbf{h}_t = \text{GRU}\left(\begin{bmatrix}\mathbf{m}_{t-1}\\ \mathbf{o}_{t-1}\end{bmatrix}, \mathbf{h}_{t-1}\right)$$

$$\mathbf{o}_t = \text{ReLU}\left(W^o \begin{bmatrix}\mathbf{a}_t\\ \mathbf{h}_t\end{bmatrix}\right)$$

$$\hat{\mathbf{y}}_t = W^C \mathbf{o}_t$$

Attention Mechanism The attention mechanism in deep learning is a fairly recent innovation [XXX] enabling the neural network to, for each unit of output, distinguish certains parts of the input as more relevant to that output, thus emulating the human tendency to focus on portions of the input, rather than dividing one's attention uniformly over the entire input (as earlier encoder-decoder arcitectures were required to do by default).

Attention can be thought of as a kind of query system, in which the last predicted unit of output can be used to 'query' the output of the encoder and create a weighted sum of the encoder (source) states for further use in the decoder.

Formally, at time step t in the decoding process,

$$\alpha_t = \text{softmax}\left(\text{ReLU}(W^e S^e + W^h \mathbf{h}^t[1]_{1\times N})\mathbf{v}\right).$$

where S^e is the matrix of encoder state vectors, $\mathbf{s}^d$ is the current decoder state vector, the W are the corresponding weight matrices, and $\mathbf{v}$ is a weight vector. The row vector $[1]_{1\times N}$ serves simply to 'copy' the single target state vector to be added to each of the encoder state vectors. We compute the attention values as a weighted sum of source states:

$$\mathbf{a}_t = \sum_{i=1}^{N} \alpha_{ti} \mathbf{s}_i^{\text{source}}.$$

Note that t is used to index decoder steps, while i indexes to encoder steps.

For convenience, the model notation used is summarized below:

- X: input consisting of N frames of $80k$-dimensional FBANK features, where k is the parameter indicating how many consicutive frames are to be stacked

- Y: $V \times M$ output matrix consisting of M V-dimensional one-hot vectors corresponding to phone IDs; V is vocabulary size

- $W^e \in \mathbb{R}^{h \times 2h}$: attention weight matrix for encoder state matrix
- $W^h \in \mathbb{R}^{h \times h}$: attention weight matrix for target state vector
- $\mathbf{h}_t$: decoder hidden state at time t
- t: decoder time step; corresponds to output phone number
- i: encoder time step; corresponds to input frames
- $\hat{\mathbf{y}}_t$: prediction for phone t
- $\mathbf{m}_t$: embedding of prediction $\hat{y}_t$ or ground truth y_t
- W^m: embedding matrix used to compute $\mathbf{m}_t$
- W^c: classifier weights, used to generate a prediction from the output vector
- a_t: attention vector at time step t
- S: matrix of encoder state targets
-
-
-
-
-
-
-

5.1.2 Articulatory Feature Models

Both the binary and contintuous articulatory feature models, (henceforth simply the binary and continuous models for brevity), retain the basic structure and approach of the phone recognition model. The primary difference is that rather than predicting a discrete sequence of phone, these models predict articulatory features rather than phones. This provides greater universality because the articulatory features allow for the description of any realizable phone[2], and the number of features is less than the phonetic inventory of nearly all langages, which allows for more compact representations.

The articulatory features are summarized in Table 1.

Encoder

Decoder

Attention Mechanism

5.2 Automatic Pronunciation Feedback

5.3 Accent Transfer in Phonetic Representation Space

Text here.

5.4 Speech Synthesis from Phonetic Representations

Text here.

[2]Some extremely rare articulatory features have been omitted due to a lack of suitable data; however the phone set used here is capable of fully describing all high-resource and nearly all medium-resource languages.

Group	Binary Feature	Continuous?	Group	Binary Feature	Continuous?
Quality	Voiced	✓	Place	Front	✓
Class	Silence	✓		Near-Front	
	Vowel	✓		Central	
	Pulmonic	✓		Near-Back	
	Pulmonic Consonant	✓		Back	
	Non-pulmonic Consonant	✓		Close	
Style	Rounded	✓		Near-Close	
	Rhoticized	✓		Close-Mid	
Manner	Plosive	✓		Open-Mid	
	Nasal	✓		Near-Open	
	Trill	✓		Open	✓
	Tap/Flap	✓		Schwa	
	Fricative	✓		Bilabial	✓
	Lateral Fricative	✓		Labiodental	✓
	Affricate	✓		Dental	✓
	Approximant	✓		Alveolar	✓
	Lateral Approximant	✓		Postalveolar	✓
	Click	✓		Retroflex	✓
	Implosive	✓		Palatal	✓
	Ejective	✓		Velar	✓
				Uvular	✓
				Pharyngeal	✓
				Glottal	✓
				Lateral	✓

Tabelle 1: Binary and Continuous Articulatory Features

6 Results

7 Conclusions

Literatur