How Accents Confound: Probing for Accent Information in End-to-End Speech Recognition Systems

Archiki Prasad
Department of EE, IIT Bombay, India
archikiprasad@gmail.com

Preethi Jyothi
Department of CSE, IIT Bombay, India
pjyothi@cse.iitb.ac.in

Abstract

In this work, we present a detailed analysis of how accent information is reflected in the internal representation of speech in an end-to-end automatic speech recognition (ASR) system. We use a state-of-the-art end-to-end ASR system, comprising convolutional and recurrent layers, that is trained on a large amount of US-accented English speech and evaluate the model on speech samples from seven different English accents. We examine the effects of accent on the internal representation using three main probing techniques: a) Gradient-based explanation methods, b) Information-theoretic measures, and c) Outputs of accent and phone classifiers. We find different accents exhibiting similar trends irrespective of the probing technique used. We also find that most accent information is encoded within the first recurrent layer, which is suggestive of how one could adapt such an end-to-end model to learn representations that are invariant to accents.

1 Introduction

Traditional automatic speech recognition (ASR) systems, consisting of independently-trained acoustic, pronunciation and language models, are increasingly being replaced by end-to-end ASR systems (Chiu et al., 2018; Hori et al., 2017). An end-to-end ASR system refers to a single model that subsumes all the traditional ASR components and directly translates a speech utterance into a sequence of graphemes. Such models benefit from jointly training acoustic and language models and eliminating the need for a pronunciation dictionary. While end-to-end ASR models have clear merits and are elegant in their formulation, they tend to be opaque in their predictions and difficult to interpret.

In order to understand better what is encoded in the layers of an end-to-end ASR system, prior work has explored the use of phone probes (classifiers) to analyze the phonetic content of representations at each layer (Belinkov and Glass, 2017; Belinkov et al., 2019). This analysis was restricted to a single accent of English. In this paper, we work with multiple accents of English and propose a number of different tools (other than phone probes) to investigate how accent information is encoded and propagated within an end-to-end ASR system.

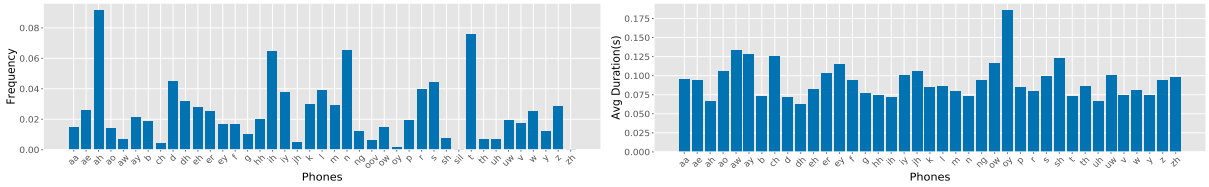
Why accented speech? We have witnessed impressive strides in ASR performance in the last few years. However, recognizing heavily accented speech still remains a challenge for state-of-the-art ASR systems. An end-to-end ASR model trained on a standard speech accent significantly underperforms when confronted with a new speech accent. To shed more light on why this happens, a systematic investigation of how such models behave when evaluated on accented speech might be useful. The insights from such an investigation might also come in handy when trying to adapt end-to-end neural architectures to be more accent-agnostic.

We tackle the following specific questions of interest in this work:

1. How do the gradients of an end-to-end ASR model behave when subject to varying accents?
2. How do we directly measure the amount of accent information encoded within hidden representations of an end-to-end model?
3. How do accents impact phone accuracy across different layers in an end-to-end model?

While the analyses of black-box models in computer vision and natural language processing have received a considerable amount of attention, prior work on the analysis of end-to-end ASR models are notably few in number. With presenting various analysis techniques that are applicable to speech,

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3739–3753
July 5 - 10, 2020.



(a) Phonetic coverage histogram

(b) Phonetic duration histogram

Figure 1: Phonetic coverage and duration histograms for the US accent. X-axis labels refer to individual phones.

we hope this work can serve as a starting point for further studies and spur more analysis-driven investigations into end-to-end ASR models. The code used in our work is publicly available.[1]

2 Experimental Setup

In this section, we first introduce the dataset of accented speech samples used in our experiments, along with details of the phone-level alignments that were necessary for our subsequent analyses. We also provide a detailed description of the specific end-to-end ASR model that we use in this work, along with important implementation details.

2.1 Dataset

We extracted accented speech samples from the Mozilla Common Voice speech corpus (Mozilla). The Voxforge corpus (Voxforge.org) was another potential source for accented speech samples. However, we preferred the Mozilla corpus as the dataset is relatively cleaner, has larger diversity in speech across accents and more importantly contains the same content rendered in different speech accents (which we exploited in our experimental analysis). We considered accented speech samples from seven different English accents: African, Australian, Canadian, England, Indian, Scotland and US. These were chosen to span the gamut of accents in terms of how they differ from the primary accent that was used to train the ASR system (US). US and Canadian serve as native accents; African, Australian and England accents are sufficiently different from the native accents while Indian and Scotland accents vary substantially.

We created a dataset of utterances in each accent using the following heuristic. First, we chose sentences that appeared in speech samples corresponding to five or more accents (including US). For African and Scotland accents that contained very few speech samples overall, we chose transcripts that had an utterance with the same text spoken by a US-accented speaker. This finally led to 3500 samples being chosen for each accent containing text that appeared in at least two accents, at most six accents and 3.24 different accents on average. We chose the utterances to largely overlap in text so that differences in ASR performance could be mostly attributed to acoustic differences and not language model-related differences.

Alignments: For our empirical investigation, we require phone alignments for all the accented speech samples. We used an existing Kaldi-based forced aligner, *gentle*[2], to align the speech samples. The aligner uses the CMU dictionary and accommodates multiple pronunciations for a word which is important for accented speech. Although the aligner was trained on US-accented speech, we found the alignments assigned to various accented speech samples to be fairly robust as determined by a manual check of the alignments for a random set of Indian-accented utterances. The aligner failed to produce outputs on samples of poor quality; these samples were omitted from our analysis.

Figure 1(a) shows the coverage across phones for the US-accented speech samples and Figure 1(b) shows the total duration of phones for US-accented speech samples. Phone coverage and phone duration distributions for all the other accents are almost identical in shape to the US accent. Aggregate plots visualizing these distributions across the remaining accents are shown in Appendix A.

2.2 End-to-end ASR: Deep Speech 2

We chose DeepSpeech2 (Amodei et al., 2016) as our end-to-end ASR model. This is a widely-used architecture that directly maps speech features to graphemes and is trained using the Connectionist Temporal Classification (CTC) loss (Graves et al.,

[1] https://github.com/archiki/ASR-Accent-Analysis/

[2] Available at https://github.com/lowerquality/gentle

2006). The input to the model is a sequence of frequency magnitude spectrograms (henceforth referred to as SPEC), obtained using a 20ms Hamming window and a stride of 10ms. With a sampling rate of 16kHz, we end up with 161-dimensional input features. The first two layers are 2D-convolutions with 32 kernels at each layer with sizes 41×11 and 21×11, respectively. Both convolutional layers have a stride of 2 in the frequency domain while the first layer and second layer have a stride of 2 and 1, respectively, in the time domain. This setting results in 1312 features per time frame after the second convolutional layer which we will henceforth refer to as CONV. The convolutional layers are followed by 5 bidirectional LSTMs (Hochreiter and Schmidhuber, 1997), each with a hidden state size of 1024 dimensions. These layers are henceforth referred to as RNN_0, RNN_1, RNN_2, RNN_3 and RNN_4. The implementation of this model is adapted from Naren (2016). This model is trained on 960 hours of US-accented speech obtained from the Librispeech corpus (Panayotov et al., 2015). All subsequent experiments use this pretrained model, which we will refer to as DS2.

Table 1 shows the performance of DS2 when evaluated on speech samples from different accents. Both word error rates (WER) and character error rates (CER) on the test sets are reported for each accent. As expected, US and Canadian-accented samples perform best.[3] DS2 has the most trouble recognizing Indian-accented samples, incurring a high WER of 49.1%, followed by Scotland-accented samples with a WER of 36.7%.

The next three sections are grouped based on the probing techniques we adopt to examine the effect of accents on the internal representations learned by the model:

- **Gradient-based analysis** of the model (§3).
- **Information-theoretic measures** to directly quantify accent information in the learned representations (§4).
- Outputs of **phone and accent classifiers** at each layer (§5).

[3]US-accented samples are drawn from various parts of the US and are more diverse in accent, compared to the Canadian-accented samples. We suspect this could be why US underperforms compared to Canada.

Accent	Utterances		Duration		Error	
	Train	Test	Train	Test	WER	CER
African	2500	1000	3	1	28.7	16.2
Australia	2500	1000	2	1	28.7	16.6
Canada	2500	1000	2	1	18.7	9.9
England	2500	1000	2	1	29.0	16.4
Indian	2500	1000	2	1	49.1	31.6
Scotland	2500	1000	2	1	36.7	22.3
US	2500	1000	3	1	20.4	10.9

Table 1: Data statistics of accented speech datasets. Duration is approximated to hours and WER/CER refer to the test error rates for each accent using DS2.

3 Gradient-based Analysis

Gradient-based techniques have been widely adopted as an explainability tool in both computer vision and NLP applications. In this section, we adapt some of these techniques to be used with speech and derive insights based on how accents modify gradient behavior.

3.1 Attribution Analysis

A simple gradient-based explanation method considers the gradient of the output f_j from a neural network (where j denotes a target class) with respect to an input x_i (where i refers to the i^{th} input time-step used to index the input sequence $\mathbf{x}$):

$$\text{grad}(j, i, \mathbf{x}) = \frac{\partial f_j}{\partial x_i}$$

Here, $\text{grad}(j, i, \mathbf{x})$ serves as an approximate measure of how much x_i contributes to f_j (Simonyan et al., 2014). For speech as input, x_i would be an acoustic feature vector (e.g. spectral features). Thus, $\text{grad}(j, i, \mathbf{x})$ would be a vector of element-wise gradients with respect to x_i. For each x_i, we use the L2 norm to reduce the gradient vectors to scalars: $a_{i,j} = \|\text{grad}(j, i, \mathbf{x})\|_2$. We refer to $a_{i,j}$ as an *attribution*. We note here that instead of using the L2 norm, one could use the dot product of the gradient $\text{grad}(j, i, \mathbf{x})$ and the input x_i as an alternate gradient-based method (Denil et al., 2014). For our task, this attribution method seemed less suitable (compared to computing the L2 norm) as dot products would have the undesirable effect of being sensitive to prosodic variations in speech and speech sounds like fricatives or stop onsets which have sparse spectral distributions. (We refer interested readers to Appendix C for visualizations using the dot product-based attribution method.)

We compute character-level attribution from the DS2 system using the following two-step approach.

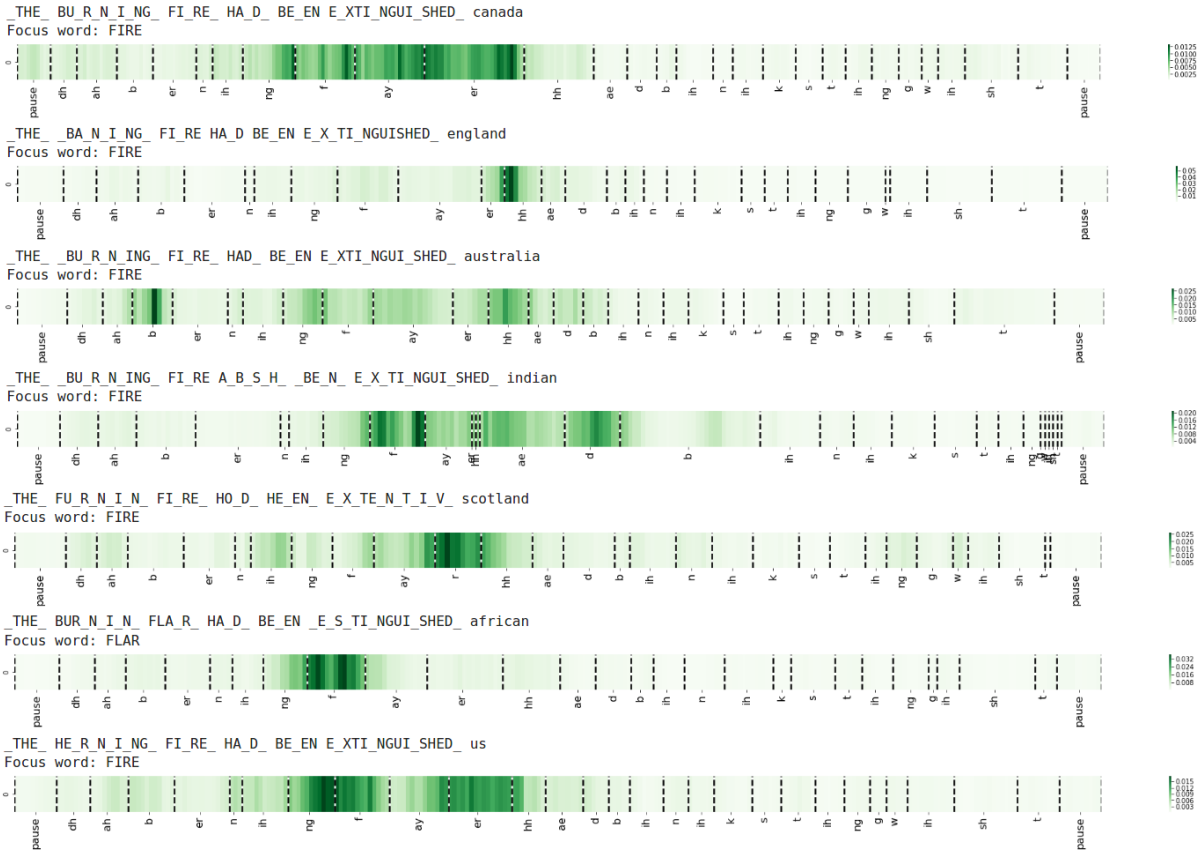


Figure 2: Example illustrating gradient attribution corresponding to the word “FIRE" across different accents.

First, we consider the output character with the highest softmax probability at each output timestep. Next, we consider only non-blank characters produced as output and sum the gradients over all contiguous repetitions of a character (that would be reduced to a single character by the CTC algorithm)[4]. Word-level attribution can be similarly computed by summing the character-level attributions corresponding to each character that makes up the word.

Figure 2 illustrates how attribution changes for a specific word, “FIRE", across different accents. We consider speech samples from all seven accents corresponding to the same underlying reference text, “The burning fire had been extinguished". Each subplot also shows the phonetic alignment of the text on its x-axis. We observe that the attributions for “FIRE" are fairly well-aligned with the underlying speech in the US and Canadian samples; the attributions appear to deviate more in their alignments for all the other accents.

To quantify the differences in alignment across accents suggested by the visualization in Figure 2, we measure the alignment accuracy using the earth mover’s distance (EMD). For each accent, we compute the EMD between two distributions, one derived from the attributions and the other from the reference phonetic alignment. The EMD between two distributions p and q over the set of frames (or rather, frame sequence numbers) T is defined as

$$\mathrm{EMD}(p, q) = \inf_{Z} \sum_{i,j \in T} |i - j| \cdot Z(i, j)$$

where the infimum is over all “transportation functions” $Z : T \times T \rightarrow \mathbb{R}_+$ such that $\sum_{j \in T} Z(i, j) = p(i)$ (for all i) and $\sum_{i \in T} Z(i, j) = q(j)$ (for all j).

Given a *correctly predicted* word, we define the distribution p as the uniform distribution over the frames that are aligned with the word, and q as the distribution obtained by normalizing the word-level attribution of the word in the utterance. For each accent, we sample a set of words that were correctly predicted (equally many for all accents) and compute the average of the EMD between the dis-

[4]The CTC algorithm produces output probabilities for observing a “blank”, signifying no label. Excluding the blank symbol from our analysis helped with reducing gradient computation time. We also confirmed that including the blank symbol did not change the results from our analysis.

Accent	EMD			
	C_0	C_1	C_2	Overall
US	43.54	42.42	39.55	42.6
Canada	42.17	39.68	40.47	40.94
Indian	53.07	47.47	49.63	50.34
African	46.63	42.61	41.05	44.3
England	47.0	41.52	43.44	44.3
Scotland	45.34	41.38	41.65	43.26
Australian	46.91	44.24	47.45	45.87

Table 2: EMD trends quantifying the difference in attributions across accents. C_0, C_1 and C_2 are clusters of words containing {1-2}, 3 and {4-5} phones, respectively

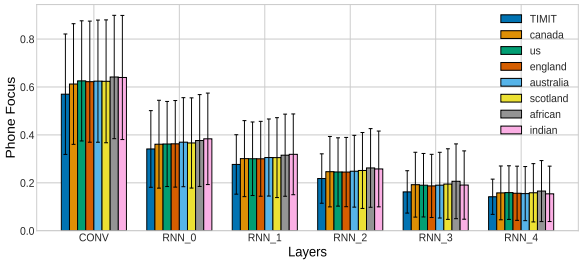


Figure 3: Comparison of phone focus across layers for various accents.

tributions p and q corresponding to each word. This average serves as an alignment accuracy measure for the accent. For the EMD analysis, we restrict ourselves to a set of 380 sentences that have corresponding speech utterances in all accents. This way, the content is mostly identical across all accents. Table 2 shows the averaged EMD values for each accent computed across all correctly predicted words. Larger EMD values signify poorer alignments. The overall values clearly show that the alignments from US and Canadian-accented samples are most accurate and the alignments from the Indian-accented samples are most inaccurate. We also cluster the words based on the number of phones in each word, with C_0, C_1 and C_2 referring to words with {1-2}, 3 and {4-5} phones, respectively. As expected, words in C_0, being smallest in size, deviate most from the reference distribution and incur larger EMD values (compared to C_1 and C_2). The overall trend across accents remains the same for each cluster.

3.2 Information Mixing Analysis

Another gradient-based analysis we carried out is to check if accents affected how, at various levels, the representation at each frame is influenced by the signal at the corresponding input frame. One can expect that, in layers higher up, the representation at each frame mixes information from more and more input frames. However, it is reasonable to expect that most of the contribution to the representation should still come from the frames in a window corresponding to the *same phone*. (We examine the contribution of neighboring phones in Appendix B)

As detailed below, we devise quantities that measure the extent of information mixing and apply them to our systems. Not surprisingly, as shown below, we do observe that mixing increases as one climbs through the layers. But somewhat surprisingly, we find that there is little variation of these trends across accents. This suggests that information mixing is largely dictated by the network itself, rather than by the details of the data.

The quantities we use to measure information mixing are inspired by Brunner et al. (2020). We define the contribution of the i^{th} input frame x_i to the final output of the network f via the representation e^l_j in a given layer l corresponding to frame j as:

$$\hat{g}^l_{i,j} = \left\| \sum_{k=1}^{d} \left(\frac{\partial f}{\partial e^l_j(k)} \right) \left(\frac{\partial e^l_j(k)}{\partial x_i} \right) \right\|_2 \quad (1)$$

where e^l_j is a d-dimensional vector ($e^l_j(k)$ refers to the k^{th} dimension of e^l_j), and f consists of the non-blank characters in the maximum probability output (after the softmax layer). We use a normalized version of $\hat{g}^l_{i,j}$ to compare the contribution to e^l_j from different x_i:

$$g^l_{i,j} = \frac{\hat{g}^l_{i,j}}{\sum_{n=1}^{T} \hat{g}^l_{n,j}}$$

For this analysis, we used a subset of 250 utterances for each accent that have almost the same underlying content.[5]

A measure of "focus" of an input phone at level l – how much the frames at level l corresponding to that phone draw their contributions from the corresponding frames in the input – is obtained by summing up $g^l_{i,j}$ over i, j corresponding to the phone. Figure 3 shows this quantity, averaged over all phones in all the utterances for each accent. We observe that the focus decreases as we move from CONV to RNN_4, with the largest drop appearing between CONV and RNN_0. This is intuitive as we expect some of the focus to shift from individual

[5]This smaller sample was chosen for faster gradient computations and gave layer-wise phone accuracies similar to what we obtained for the complete test set of 1000 utterances. A plot showing these consistent trends is included in Appendix D

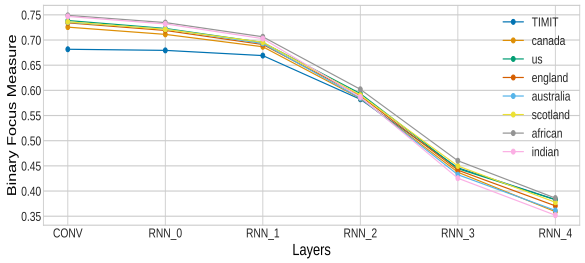


Figure 4: Variation in binary focus measure, averaged over all the phones, across layers for various accents.

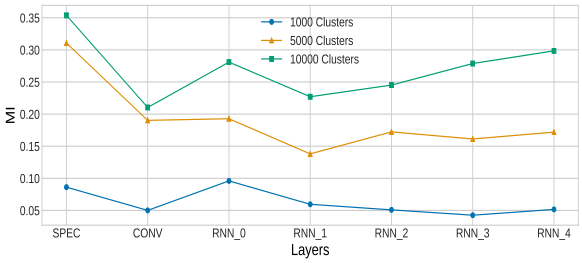


Figure 5: Mutual Information between hidden representations and accents across layers.

phones to their surrounding context, as we move to a recurrent layer (from the CONV layer). This trend persists in moving from RNN_0 to RNN_4 with the focus on individual phones steadily dropping. We also see a consistent albeit marginal trend across accents with US/Canadian-accented samples showing the lowest focus.

For each input phone, one can also define a binary measure of focus at level l, which checks that the focus of the frames at that level has not shifted to an input phone other than the one whose frames it corresponds to. That is, this binary focus measure is 1 if the focus of the phone at a level as defined above is larger than the contribution from the input frames of every *other* phone. Figure 4 shows how this measure, averaged across all phones for each accent, varies across layers. Again, we see that focus is highest in the first CONV layer, dropping to 70% at RNN_1 and 45% at RNN_3. Further, again, we observe very similar trends across all accents.

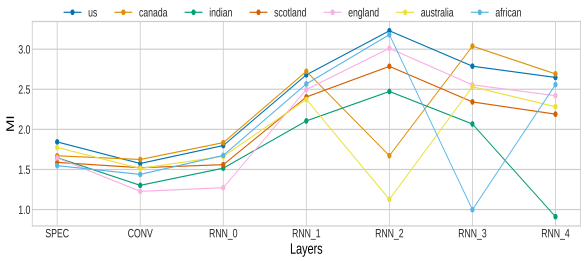
From both the above analyses of focus, we observe that there is a pronounced drop in focus through the layers, but this trend is largely independent of the accent. We also plot variations for the well-known TIMIT database (Garofolo, 1993) in both Figures 3 and 4 to confirm that the same trend persists. For TIMIT, we used the samples from the specified test set along with the phonetic alignments that come with the dataset. We conclude that information mixing, and in particular, the measures of focus we used, are more a feature of the network than the data.

4 Information-Theoretic Analysis

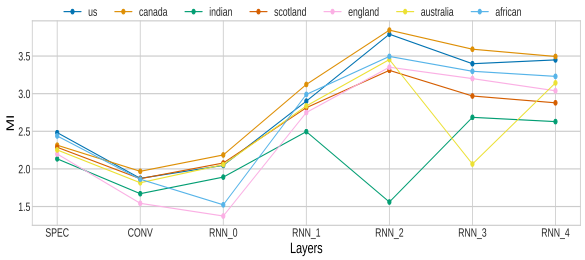
In the previous section, we used gradient-based methods to examine how much an input frame (or a set of frames corresponding to a phone or a word) contributes to the output and how these measures change with varying accents. Without computing gradients, one could also directly measure how much information about accents is encoded within the representations at each layer. Towards this, motivated by Voita et al. (2019), we compute the mutual information (MI) between random variables $e^l_{\mathbf{x}}$ and α, where $e^l_{\mathbf{x}}$ refers to a representation at layer l corresponding to input $\mathbf{x}$ and $\alpha \in [0, 6]$ is a discrete random variable signifying accents. We define a probability distribution for $e^l_{\mathbf{x}}$ by discretizing the space of embeddings via k-means clustering (Sajjadi et al., 2018). We use mini-batched k-means to cluster all the representations corresponding to files in the test sets mentioned in Table 1 across accents and use the cluster labels thereafter to compute MI.

Figure 5 shows how MI varies across different layers for three different values of k. Increasing k would naturally result in larger MI values. (The maximum possible value of MI for this task would be $\log_2(7)$.) We observe a dip in MI going from spectral features SPEC to CONV, which is natural considering that unprocessed acoustic features would contain most information about the underlying accent. Interestingly, we observe a rise in MI going from CONV to RNN_0 signifying that the first layer of RNN-based representations carries the most information about accent (not considering the acoustic features). All subsequent RNN layers yield lower MI values.

Apart from the MI between representations and accents that capture how much accent information is encoded within the hidden representations, we also compute MI between representations and a discrete random variable signifying phones. The MI computation is analogous to what we did for accents. We will now have a separate MI plot across layers corresponding to each accent. Figure 6 shows the MI values across layers for each accent when $k = 500$ and $k = 2000$. We see an overall trend of increasing MI from initial to later layers. Interestingly, the MI values across ac-



(a) Cluster Size: 500



(b) Cluster Size: 2000

Figure 6: Mutual Information between representations and phones for different clusters sizes and accents.

cents at RNN_4 exhibit a familiar ordering where US/Canadian accents receive the highest MI value while Indian and Scotland's accents receive the lowest MI value.

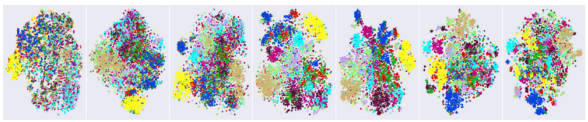
We also attempt to visualize the learned phone representations by projecting down to 2D. For a specific phone, we use the precomputed alignments to compute averaged layer-wise representations across the frames within each phone alignment. Figure 7 shows t-SNE based (Maaten and Hinton, 2008) 2D visualizations of representations for the 10 most frequent phones in our data, {*'ah', 'ih', 'iy', 'dh', 'd', 'l', 'n', 'r', 's', 't'*}. Each subplot corresponds to a layer in the network. The plots for phones from the US-accented samples appear to have slightly more well-formed clusters, compared to the Indian-accented samples. These kinds of visualizations of representations are, however, limiting and thus motivates the need for analysis like the MI computation presented earlier.

5 Classifier-driven Analysis

5.1 Accent Classifiers

We train an accent classifier for each layer that takes the corresponding representations from the layer as its input. We implemented a classifier with two convolutional layers of kernel size, stride and padding set to (31,21), (3,2), (15,10) and (11,5), (2,1) and (5,2), respectively. We used batch normalization (Ioffe and Szegedy, 2015) followed by ReLU activations for each unit. The network also contained two max-pooling layers of size (5,3) and (3,2), respectively, and a final linear layer with hidden dimensionality of 500 (with a dropout rate of 0.4). Table 1 lists the number of utterances we used for each accent for training and evaluation. The accent classifiers were trained for 25 epochs using Adam optimizer (Kingma and Ba, 2015) and a learning rate of 0.001.

(a) US accent



(b) Indian accent

Figure 7: t-SNE plot for representations of top 10 phones across US and Indian-accented samples, with the following layers on the X-axis (from left to right): SPEC, CONV, RNN_0, RNN_1, RNN_2, RNN_3, RNN_4 .

Figure 8 shows the accent accuracies obtained by the accent classifier specific to each layer (along with error bars computed over five different runs). RNN_0 is most accurate with an accuracy of about 33% and RNN_4 is least accurate. It is interesting that RNN_0 representations are most discriminative across accents; this is also consistent with what we observe in the MI plots in Figure 5.

5.2 Phone Classifiers

Akin to accent classifiers, we build a phone classifier for each layer whose input representations are labeled using phone alignments. We train a simple multi-layer perceptron for each DS2 layer (500-dimensional, dropout rate of 0.4) for 10 epochs

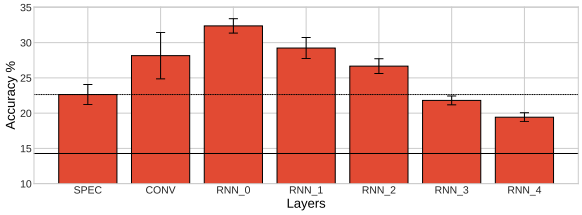


Figure 8: Accuracy (%) of accent probes trained on hidden representations at different layers.

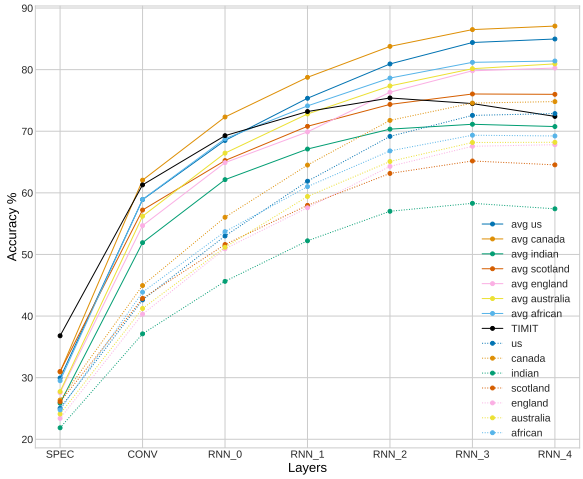


Figure 9: Trends in accuracy (%) of phone probes for frame-level (dotted) and averaged representations (solid) at different layers.

using the Adam optimizer. We train both frame-level classifiers, as well as phone-level classifiers that use averaged representations for each phone as input. The accuracies of both types of phone classifiers are shown in Figure 9. As expected, the phone accuracies improve going from SPEC to RNN_4 and the accuracies of US/Canadian samples are much higher than that of Indian samples. Classifiers using the averaged representations consistently perform much better than their frame-level counterparts. We note that Belinkov and Glass (2017) report a dip in phone accuracies for the last RNN layers, which we do not observe in our experiments. To resolve this inconsistency, we ran phone classifiers on TIMIT (which was used in Belinkov and Glass (2017)) using representations from our DS2 model and the dip in RNN_4 accuracies surfaced (as shown in Figure 9). This points to differences between the TIMIT and Mozilla Common Voice datasets. (An additional experiment examining how phone classifiers behave on different datasets is detailed in Appendix D.)

6 Discussion

This is the first detailed investigation of how accent information is reflected in the internal representations of an end-to-end ASR system. In devising analysis techniques for ASR, while we do follow the broad approaches in the literature, the details are often different. Most notably, the use of EMD for attribution analysis is novel, and could be of interest to others working with speech and other temporal data. Similarly, the phone focus measures in the information mixing analysis are new. We also highlight that this is the first instance of analysis of ASR consisting of multiple analysis techniques. On the one hand, this has uncovered robust trends that manifest in more than one analysis. On the other hand, it also shows how some trends are influenced more by the neural-network architecture more than the data. This provides a platform for future work in speech neural-network analysis, across architectures, data-sets and tasks.

In our results, we encountered some unexpected details. For instance, while the RNN_0 layer is seen to reduce the phone focus the most, uniformly across all accents (as shown in Figure 3, it is also seen to segregate accent information the most, recovering accent information "lost" in the convolution layer (as shown in Figure 5). We also see this trend surfacing in Figure 8 where the accent classifier gives the highest accuracy for RNN_0 and the accuracies quickly taper off for subsequent layers. This suggests that the first RNN layer is most discriminative of accents. Models that use an adversarial objective to force the representations to be accent invariant (e.g., (Sun et al., 2018)) might benefit from defining the adversarial loss as a function of the representations in the first RNN layer.

7 Related Work

7.1 Accented Speech Recognition

Huang et al. (2001) show that accents are the primary source of speaker variability. This poses a real-world challenge to ASR models which are primarily trained on native accented datasets. The effect of accents is not limited to the English language, but also abundant in other languages such as Mandarin, Spanish, etc.

An interesting line of work exploits the ability to identify accents in order to improve performance. Zheng et al. (2005) combine accent detection, accent discriminative acoustic features, acoustic model adaptation using MAP/MLLR and model selection to achieve improvements over accented Mandarin speech.Vergyri et al. (2010) investigate the effect of multiple accents on the performance of an English broadcast news recognition system using a multiple accented English dataset. They report improvements by including data from all accents for an accent-independent acoustic model training.

Sun et al. (2018) propose the use of domain adversarial training (DAT) with a Time Delay Neu-

ral Network (TDNN)-based acoustic model. They use native speech as the source domain and accented speech as the target domain, with the goal of generating accent-invariant features which can be used for recognition. Jain et al. (2018) also use an accent classifier in conjunction with a multi-accent TDNN based acoustic model in a multi-task learning (MTL) framework. Further, Viglino et al. (2019) extended the MTL framework to use an end-to-end model based on the DS2 architecture and added a secondary accent classifier that uses representations from intermediate recurrent layers as input. Chen et al. (2020) propose an alternate approach using generative adversarial networks (GANs) to disentangle accent-specific and accent-invariant components from the acoustic features.

7.2 Analysis of ASR Models

Nagamine et al. (2015, 2016) were the first to examine representations of a DNN-based acoustic model trained to predict phones. They computed selectivity metrics for each phoneme and found better selectivity and more significance in deeper layers. This analysis was, however, restricted to the acoustic model. Belinkov and Glass (2017) were the first to analyze a Deep Speech 2 model by training phone classifiers that used representations at each layer as its input. These ideas were further extended in Belinkov et al. (2019) with classifiers used to predict phonemes, graphemes and articulatory features such as place and manner of articulation. Belinkov and Glass (2019) present a comparison of different analysis methods that have been used in prior work for speech and language. The methods include recording activations of pretrained networks on linguistically annotated datasets, using probing classifiers, analyzing attention weights and ABX discrimination tasks (Schatz et al., 2013).

Other related work includes the analysis of an audio-visual model for recognition in Alishahi et al. (2017), where the authors analyzed the activations of hidden layers for phonological information and observed a hierarchical clustering of the activations. Elloumi et al. (2018) use auxiliary classifiers to predict the underlying style of speech as being spontaneous or non-spontaneous and as having a native or non-native accent; their main task was to predict the performance of an ASR system on unseen broadcast programs. Analogous to saliency maps used to analyze images, Li et al. (2020) propose reconstructing speech from the hidden representations at each layer using highway networks. Apart from ASR, analysis techniques have also been used with speaker embeddings for the task of speaker recognition (Wang et al., 2017).

The predominant tool of choice for analyzing ASR models in prior work has been classifiers that are trained to predict various phonological attributes using quantities extracted from the model as its input. We propose a number of alternatives other than just the use of classifiers to probe for information within an end-to-end ASR model. We hope this spurs more analysis-driven investigations into end-to-end ASR models.

8 Summary

This work presents a thorough analysis of how accent information manifests within an end-to-end ASR system. The insights we gleaned from this investigation provide hints on how we could potentially adapt such end-to-end ASR models, using auxiliary losses, to be robust to variations across accents. We will investigate this direction in future work.

Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback and comments. The second author gratefully acknowledges support from a Google Faculty Research Award and IBM Research, India (specifically the IBM AI Horizon Networks-IIT Bombay initiative).

References

Afra Alishahi, Marie Barking, and Grzegorz Chrupała. 2017. Encoding of phonology in a recurrent neural model of grounded speech. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 368–378.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In Proceedings of the 33rd International Conference on Machine Learning, pages 173–182.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. Analyzing Phonetic and Graphemic Representations in End-to-End Automatic Speech Recognition. In Proc. Interspeech 2019, pages 81–85.

Yonatan Belinkov and James Glass. 2017. Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems. In Advances in Neural Information Processing Systems, pages 2441–2451.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. Transactions of the Association for Computational Linguistics, pages 49–72.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On Identifiability in Transformers. In International Conference on Learning Representations.

Yi-Chen Chen, Zhaojun Yang, Ching-Feng Yeh, Mahaveer Jain, and Michael L Seltzer. 2020. AIPNet: Generative Adversarial Pre-training of Accent-invariant Networks for End-to-end Speech Recognition. In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6979–6983.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4774–4778.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of Salient Sentences from Labelled Documents. arXiv preprint arXiv:1412.6815.

Zied Elloumi, Laurent Besacier, Olivier Galibert, and Benjamin Lecouteux. 2018. Analyzing Learned Representations of a Deep ASR Performance Prediction Model. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 9–15.

John S Garofolo. 1993. Timit acoustic phonetic continuous speech corpus. Linguistic Data Consortium, 1993.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the 23rd International Conference on Machine Learning, pages 369–376.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural computation, 9(8):1735–1780.

Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. 2017. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In Proc. Interspeech 2017, pages 949–953.

Chao Huang, Tao Chen, Stan Li, Eric Chang, and Jianlai Zhou. 2001. Analysis of speaker variability. In Seventh European Conference on Speech Communication and Technology, pages 1377–1380.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, pages 448–456.

Abhinav Jain, Minali Upreti, and Preethi Jyothi. 2018. Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning. In Proc. Interspeech 2018, pages 2454–2458.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).

Chung-Yi Li, Pei-Chieh Yuan, and Hung-Yi Lee. 2020. What Does a Network Layer Hear? Analyzing Hidden Representations of End-to-End ASR Through Speech Synthesis. In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6434–6438.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research, 9:2579–2605.

Mozilla. Mozilla common voice dataset. https://voice.mozilla.org/en/datasets.

Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. 2015. Exploring How Deep Neural Networks Form Phonemic Categories. In Proc. Interspeech 2015, pages 1912–1916.

Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. 2016. On the Role of Nonlinear Transformations in Deep Neural Network Acoustic Models. In Proc. Interspeech 2016, pages 803–807.

Sean Naren. 2016. https://github.com/SeanNaren/deepspeech.pytorch.git.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5206–5210.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. 2018. Assessing Generative Models via Precision and Recall. In Advances in Neural Information Processing Systems, pages 5228–5237.

Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux. 2013. Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. In Proc. Interspeech 2013, pages 1–5.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In International Conference on Learning Representations, ICLR, Workshop Track Proceedings.

Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie. 2018. Domain adversarial training for accented speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4854–4858.

Dimitra Vergyri, Lori Lamel, and Jean-Luc Gauvain. 2010. Automatic Speech Recognition of Multiple Accented English Data. In Proc. Interspeech 2010, pages 1652–1655.

Thibault Viglino, Petr Motlicek, and Milos Cernak. 2019. End-to-end Accented Speech Recognition. Proc. Interspeech 2019, pages 2140–2144.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4396–4406.

Voxforge.org. Free and open source speech recognition (linux, windows and mac) - voxforge.org. http://www.voxforge.org/. Accessed 06/25/2014.

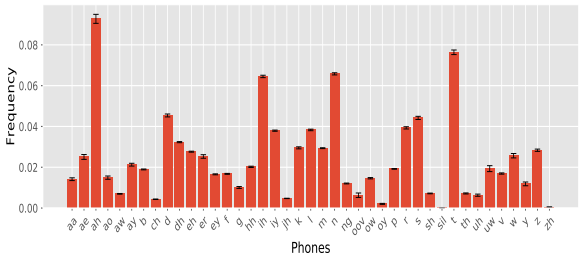
Shuai Wang, Yanmin Qian, and Kai Yu. 2017. What does the speaker embedding encode? In Proc. Interspeech 2017, pages 1497–1501.

Yanli Zheng, Richard Sproat, Liang Gu, Izhak Shafran, Haolang Zhou, Yi Su, Daniel Jurafsky, Rebecca Starr, and Su-Youn Yoon. 2005. Accent detection and speech recognition for shanghai-accented mandarin. In Ninth European Conference on Speech Communication and Technology, pages 217–220.

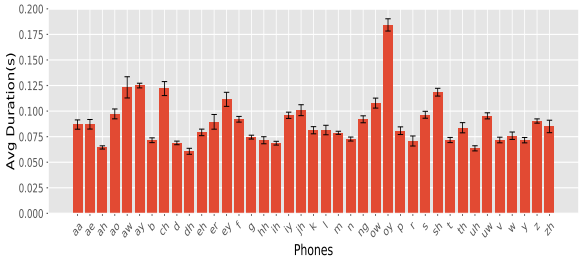
Appendix

A Dataset Information across Accents

Figure 10(a) shows the frequency of each phone across all the accents used in our dataset. Figure 10(b) shows the average duration in seconds of each phone across all the accents in our dataset. The error bars for each phone denote the variance in coverage and duration across all the accents. We observe that the variance is very small, thus indicating that the difference in phone coverage and duration across accents is minimal.



(a) Phonetic coverage histogram



(b) Phonetic duration histogram

Figure 10: Histograms showing phonetic coverage and duration for all accents with labels on the X-axis showing phones.

B Information Mixing: Neighbourhood Analysis

We explore the variation in the phone focus measure described in Section 3.2 for the aligned phone and its neighbors that precede and succeed it, across different layers of the model. Figure 11 shows that the focus of the actual (input) phone is maximum for the CONV layer and shows the fastest decrease across neighbors. This is expected due to the localized nature of convolutions. From RNN_0 to RNN_4 the focus of the actual phone decreases and is increasingly comparable to the first (and other) neighbors. We see an increase in neighbors 12^{th} and onwards because of its cumulative nature. Figure 11 shows this analysis for the TIMIT dataset. We found the trends from such a comparison of phone focus across neighbors on our accented datasets to be very similar to the trends exhibited by the different layers on TIMIT.

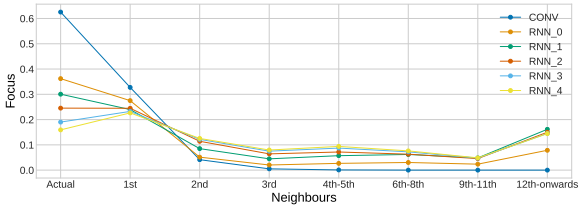


Figure 11: Phone focus of the aligned (actual) phone as compared to its preceding and succeeding neighbors on the TIMIT dataset.

C Experiments on Attribution Analysis

Common gradient-based explainability techniques include examining the gradient, as well as the dot product of the gradient and the input. We analyze both these techniques in this section. We also compare grapheme-level attributions with word-level attributions.

In Figure 12, we visualize grapheme-level attributions for the text “I’m going to them". The grapheme-level attribution is shown for the first letter in the transcription. The blue heatmaps correspond to computing the absolute value of the dot product of the gradient and the input (referred to as INP-GRAD)[6] and the green heatmaps correspond to computing the L2 norm of the gradient (referred to as GRAD). On comparing the two, we observe that the former is more diffuse and discontinuous than the latter. In general, we observe that the grapheme-level attributions are distributed non-uniformly across the frames of the underlying phone. For some accents, the attribution of the frames of the nearby phones is also comparable.

Figure 13 shows the word-level attributions for the word “FIRE" using INP-GRAD. This can be contrasted with the word-level attributions for the same word shown in Figure 2 in Section 3.1. There is more discontinuity in INP-GRAD compared to GRAD; this could be attributed to the underlying speech containing sparse spectral distributions near fricatives or stop onsets, thus making alignments from the former technique less reliable for further downstream processing.

[6]Unlike tasks like sentiment analysis where the positive and negative signs of the dot product carry meaningful information, in our setting we make use of the absolute value of the dot product.

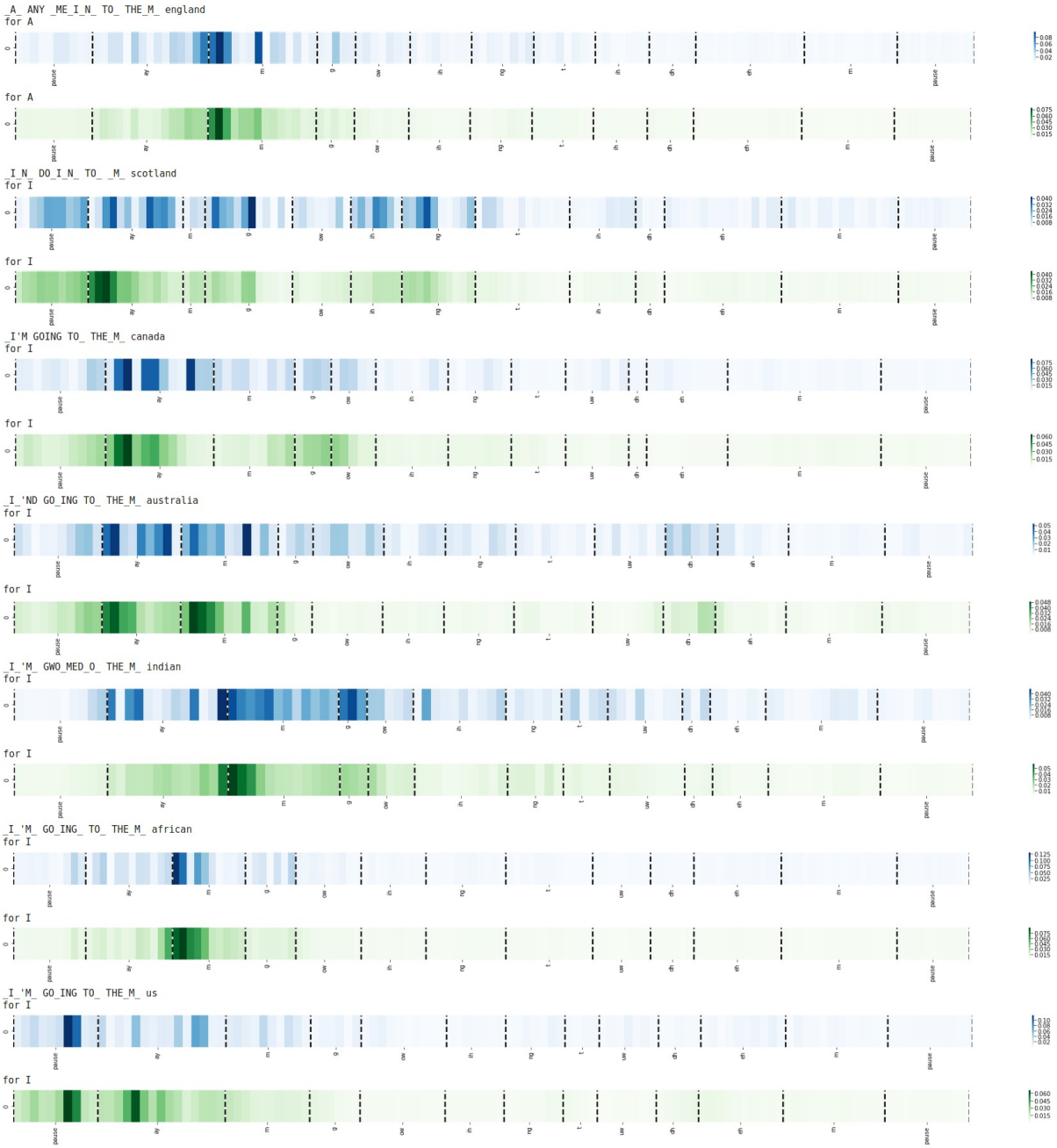


Figure 12: Grapheme-level attributions for the first letter in the text transcribed by the model.

D Phone Classifiers

D.1 Effect of Changing Distribution of Phones

We investigate the influence of changing the distribution of phones on phone classifier accuracies. We sample phones from the Mozilla Common Voice dataset so as to mimic the phone distribution of the TIMIT dataset. Figure 14 shows no significant difference in changing the phone distribution. The plot also shows the accuracy on the TIMIT dataset which is higher than the phone accuracies for the speech samples from Mozilla Common Voice for all layers (except RNN_3 and RNN_4). This reflects the differences in both datasets; TIMIT comprises clean broadband recordings while the speech samples from Common Voice are much noisier.

D.2 Effect of Changing Sample Size

In Section 3.2, we subsample 250 utterances from 1000 in the original test set. Even with only 250 utterances, we find the phone accuracy trends to be preserved as shown in Figure 15.

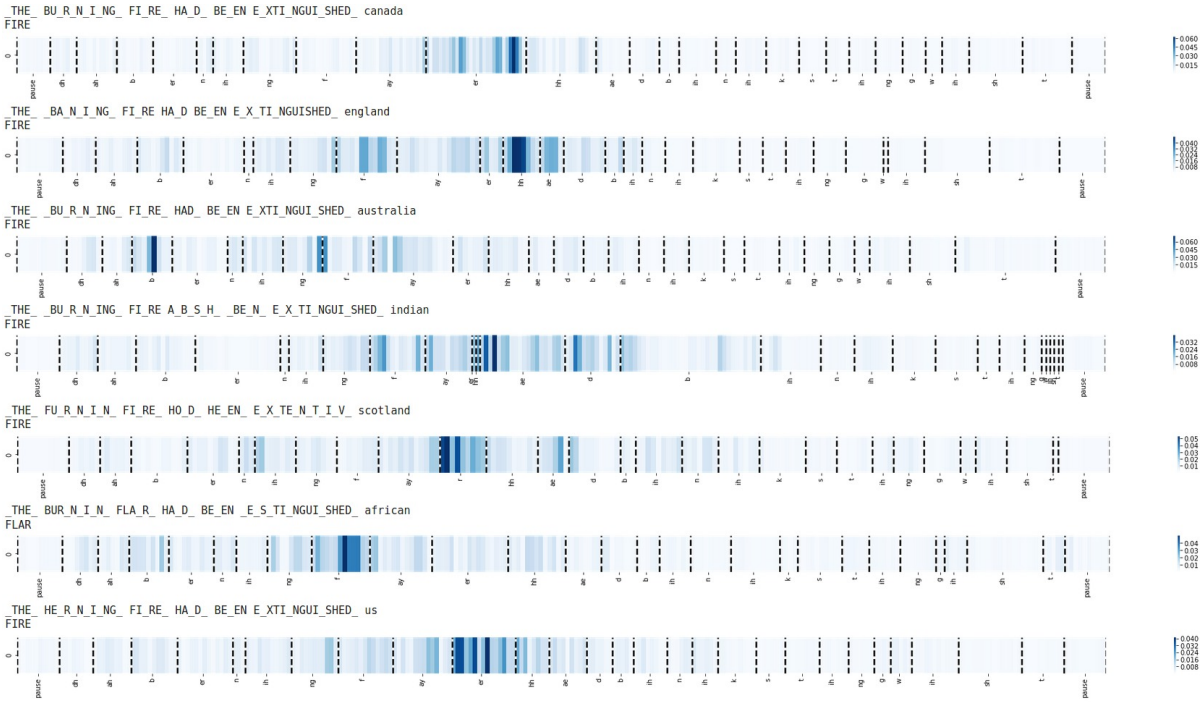


Figure 13: Word-level attributions for the word corresponding to "FIRE" in the transcription by the model.

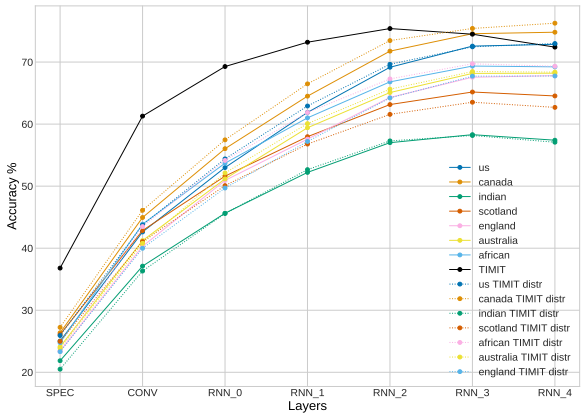


Figure 14: Trends in phone accuracy on Common Voice accented speech samples using TIMIT's phone distribution (dotted) and the original phone distribution (solid). The line in black shows performance on the TIMIT dataset.

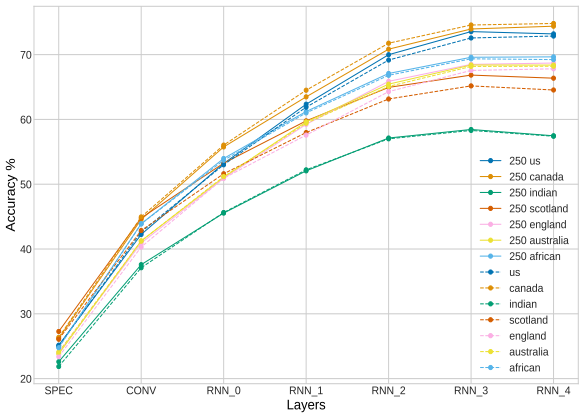


Figure 15: Trends in phone accuracy for 250 utterances (solid) randomly sampled from our test set consisting of 1000 utterances (dashed) for each accent.

D.3 Confusion in Phones

We analyze the confusion in the phones for each accent[7]. As expected, phone confusions for each accent are more prevalent in initial layers compared to the later layers. A comparison between the confusion matrices at layer RNN_4 for all accents shows a prominent difference between native accents, like US and Canada, and non-native accents, like Indian and Scotland, indicating that for the latter even at the final layers there is some residual confusion remaining. Instead of showing all the confusion matrices for each accent, we resort to an aggregate entropy-based analysis. In Figure 16, we compute the entropy of the phone distributions, averaged over all phones, across layers for different accents. We observe that in the beginning the accents are clustered together and they diverge gradually as we move to higher layers. As we approach the last 3 recurrent layers (RNN_2, RNN_3 and RNN_4), we find a clear separation, with Indian followed by Scotland accent having the highest entropy while

[7]Available at https://github.com/archiki/ASR-Accent-Analysis/tree/master/PhoneProbes

Accent	Top-5 Confusions				
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
US	*aa-ao* (0.134)	*zh-jh* (0.125)	*z-s* (0.112)	*aa-ah* (0.109)	*g-k* (0.092)
Canada	*zh-ah* (0.118)	*zh-v* (0.088)	*g-k* (0.074)	*th-t* (0.072)	*y-uw* (0.070)
African	*z-s* (0.102)	*ae-ah* (0.098)	*er-ah* (0.098)	*aa-ah* (0.089)	*g-k* (0.086)
Australia	*zh-sh* (0.462)	*th-t* (0.115)	*aa-ah* (0.105)	*g-k* (0.097)	*z-s* (0.92)
England	*zh-sh* (0.222)	*z-s* (0.126)	*aa-ah* (0.121)	*ae-ah* (0.119)	*th-t* (0.114)
Scotland	*jh-d* (0.148)	*ch-t* (0.146)	*zh-sh* (0.125)	*zh-ey* (0.125)	*ae-ah* (0.123)
Indian	*zh-ah* (0.38)	*th-t* (0.262)	*z-s* (0.175)	*uh-uw* (0.154)	*er-ah* (0.153)

Table 3: Five highest confusion pairs for all accents. Notation: phone$_i$-phone$_j$ (x) means that phone$_i$ is confused with phone$_j$ with likelihood x.

US and Canada samples have the lowest entropy.

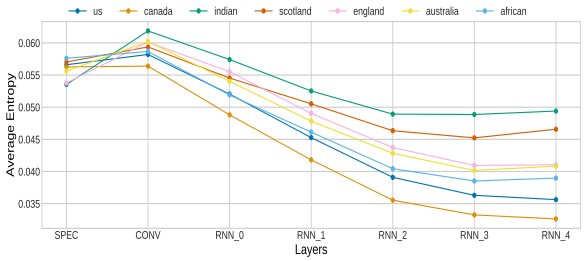


Figure 16: Entropy of the distribution of the outputs of the phone probes averaged across phones and plotted for each accent across layers.

Table 3 shows the five highest confusion pairs for each accent. We observe that each accent displays a different trend and most confusions make phonetic sense; for example, *z* getting confused as *s*, or *g* getting confused as *k*.