BeConfident: Differences in Fluent Speakers vs New Language Learners

Annie Zhu
anniezhu@stanford.edu

Khaing Mon
ksmon@stanford.edu

Gui David
gdavid@stanford.edu

Abstract

We conduct an exploratory analysis of the differences between native English speakers and new English learners in hopes of finding common speech trends among the learners. In doing so, we create a pipeline to extract embeddings from different layers of the Hubert model and test out various clustering methods such as UMAP and k-means. In our experiments, we experiment with three different layers of a Hubert Model. As a baseline, we were able to find distinct pronunciation differences between Native English Speakers and English Learners across all three layers of the model. Next, we found through experimenting with different UMAP parameters that using cosine similarity and constraining the size of the local neighborhood during dimension reduction led to distinct clusters across all layers of the model, with the last layer embeddings proving to be the most informative. Lastly, we use cluster validation methods to affirm that dimension reduction and feature examination are extremely important in clustering pronunciation differences.

1 Introduction

We are collaborating with BeConfident, a company that allows language learners to interact with AI agents through multiple platforms for personalized language lessons. Students are able to practice pronunciation and listening skills by interacting with the AI tutor, who can respond and provide feedback live on a call. BeConfident data allows us access to audio messages straight from English learners as they interact with their AI tutors. In this paper, we explore different embeddings and clustering methods to see if we can 1) find salient differences between native and non-native pronunciations and 2) gain meaningful insights on learners' speech patterns through clusters.

2 Literature Review

Comparative analyses, such as those conducted by Pasad et al. 2023, highlight the importance of examining embeddings across different layers of speech models to capture nuanced variations. Clustering these embeddings to identify meaningful patterns requires effective dimensionality reduction and UMAP, as discussed by McInnes et al. 2020, is a powerful tool for that. Additionally, the extensive comparative study by Arbelaitz et al. 2013 on cluster validity indices underscores the necessity of employing rigorous validation metrics to assess the quality of clusters. Furthermore, Bartelds and Wieling 2022 tackle the problem of quantifying language variation in Dutch regional dialects using deep acoustic models. They demonstrate how models like wav2vec 2.0 can effectively distinguish dialects with minimal data, outperforming traditional phonetic transcription methods. This study highlights the potential of leveraging hidden layers of acoustic models for fine-grained linguistic analysis in low-resource environments. However, we find a gap in the literature for the exact task of clustering mispronunciations.

3 Dataset

All the audio recordings used for training were provided by BeConfident. Across the 100 audio recordings of full utterances they provided, we found 86 different mispronunciations of single words – some examples include words "restaurants" and "plastered." We identified these errors manually while using the pronunciation scores from the Microsoft API as guidance. Since we were given the raw transcripts as part of our data, we used Montreal Forced Alignment (MFA) to create a time-aligned transcription. The MFA provided us with a .textgrid file that specified the exact time frame at which a single word began and ended. From there, we segmented the words of interest. From this, all

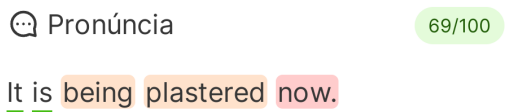

Figure 1: A datapoint and its pronunciation score

```
intervals [7]:
    xmin = 2.49
    xmax = 3.03
    text = "plastered"
intervals [8]:
    xmin = 3.03
    xmax = 3.5
    text = "now"
intervals [9]:
    xmin = 3.5
    xmax = 4.117833
    text = ""
```

Figure 2: Parsed TextGrid file of datapoint

group members recorded themselves saying these 86 words. Below is an example of a piece of data that was provided: As seen in Figure 1, we were able to determine that the word "now" was being mispronounced. Qualitatively, "now" sounded like "bow" in the user's audio clip. From there, we used MFA to segment the exact "now" timeframe (Figure 2). We then segmented the audio clip and all three members recorded themselves say the same word as well.

These recordings were then regarded as fluent-speaker recordings. Before obtaining the difference embeddings, two members combined their embeddings to get an average fluent speaker embedding and one member was regarded as the control fluent speaker embedding. For privacy reasons, we will not be releasing the entire dataset.

TODO: more detail on examples? should i explain why I cannot link the recordings?

4 Baseline Methods

To obtain embeddings that represent each audio clip, we extract the representations from the last hidden Transformer layer of the deep acoustic Hubert model. To normalize across words, we look at the difference embedding constructed by subtracting the users embedding from the embedding created by the fluent speakers saying the same word. Then, we cluster these difference embeddings using two techniques: First, by using the UMAP reduction to visualize the clusters. Second, by using k-means in tandem with cluster validity indices to identify the most appropriate number of clusters. Some of our code can be found here.

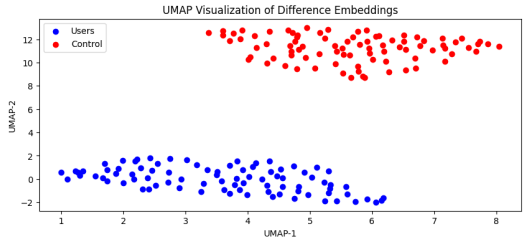

Figure 3: All UMAP parameters are at default values.

4.1 Acoustic Models

Though strong results were found using Wav2vec2 model in this paper by Bartelds and Wieling(2022), we used the Hubert model (Hsu et al., 2021) for our initial analysis because the Hubert model does a better job at capturing phonetic differences according to previous research (Pasad et al., 2023). The exact model that we used was the Hubert-Base for Keyword Spotting. This model is trained on a base model called Hubert-base-ls960, which was pretrained entirely on 16khz sampled speech. The base model comes with 95M parameters and 12 transformer layers.

4.2 Pronunciation Differences

As a baseline finding, we see that the user embeddings have a larger magnitude than the control embeddings as shown in the Figure 3, and this suggests that the embeddings even when reduced to two dimensions captured information about pronunciation differences by English-learners.

4.3 Clustering

The first method of clustering used was UMAP, the Uniform Manifold Approximation and Projection for Dimension Reduction, which projects our data into a lower dimension. The initial graph produced using UMAP did not show any meaningful clusters so to iterate, we experimented with changing the parameter values: n_neighbors, min_dist, and metric. The second method explored was k-means clustering on the difference embeddings without dimension reduction. The biggest challenge with using k-means is determining the number of clusters. To compare different clusterings against each other we use cluster validity indices.

4.4 Validation

For evaluation, we used two metrics: Davies-Bouldin and Calinski-Harabasz. These two metrics were found to be statistically significant (Arbelaitz et al., 2013). Both metrics are internal evaluation

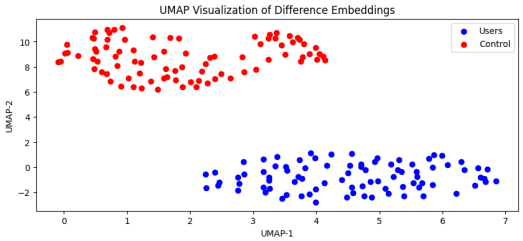

Figure 4: 2nd Layer Embeddings UMAP

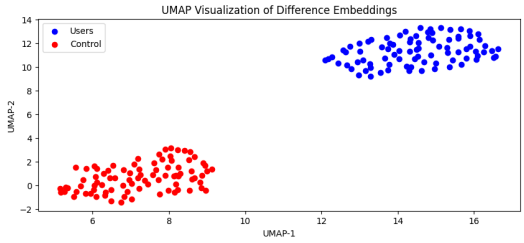

Figure 5: 11th Layer Embeddings UMAP

schemes, where a cluster is quantified as being better if points from the same cluster are close together and points from different clusters are far apart.

5 Iterative Methods

5.1 More Model Layers

We explored how the experiments above would yield different results from different layers of the Hubert model. As seen in the literature review, different layers of a transformer seem to provide different nuanced embeddings of the audio clip. For this reason, we decided to run the experiments on layers 2, 11, and last layers of the model.

First, we conduct the baseline experiments and reduce the dimensions of the user and control embeddings to 2 to see if there are salient differences between the two groups. Across all the layers, I was able to find separate clusters between the user vs. control embeddings (Figures 4, 5, 3 provide the results for each layer, respectively). This is done using only UMAP and all its default parameters. This further supports the claim that our baseline hypothesis is true – that there are notable pronunciation differences in the embedding space. It was interesting, however, how the order and the magnitude of the embeddings did not remain consistent throughout the different layers. This indicates that the embeddings from each layer are different and nuanced, but that there are still notable differences found between the user vs. control across all the layers. Since the baseline experiments showed that all layers showed differences between user and control embeddings, we proceeded to run all of the clustering experiments on all three layers. The results are further discussed in the "Results and Discussion" section.

5.2 Insights with Phonetics

talk about annie's work

5.3 Scaling to More Data

talk about gui's work

6 Results and Discussion

6.1 UMAP Metric

We were able to find through experimenting with the UMAP parameters that distinct clusters formed when the "metric" was set as cosine. We predict that the cosine metric is better than the Euclidean metric because ignoring the magnitude of the vectors to only look at the angle of the vectors allows more patterns to emerge. The plots of the layers (in order) can be seen in Figures 6, 7, and 8. Note that in all layers, we can see that five distinct clusters are formed when using the cosine metric.

6.2 UMAP N_Neighbors

Experimenting with UMAP's n_neighbors parameter also revealed insights about our embeddings. N_Neighbors controls the size of the local neighborhood of clusters, which ultimately allows the algorithm to focus either on local or global structures within the data. Consequently, a small n_neighbors value indicates a high concentration of very local structures to form clusters. The plots of the layers (in order) include Figures 10,11, and 9. Here, we see that clusters formed became more and more distinct as the number of transformer layers increased. We can see in Figure 9 that 7 distinct clusters were formed. This suggested to us that (1) focusing on local structures helps us find clusters and (2) the last layer embeddings would be the most helpful to us for further analysis.

Then, we manually assessed the quality of these clusters in 9 by re-listening to the recordings. We found that it was easier to understand the smaller clusters. For example, in Figure 9, Cluster 6 seemed to represent places where the speaker was over-emphasizing the "ə" sound and missing another constant sound; however, it was very difficult to understand what Cluster 0 represented as it was

a large group with a wide range of pronunciation mistakes.

6.3 Cluster Validity

Next, we have the k-means work with our cluster validity indices. For the Calinski index, the higher the value, the better the clustering. In our case, we saw that the clustering only gets worse and worse when the number of clusters increases (12). This behavior suggests that there does not exist a strong inherent clustering that can be captured latently by k-means. **do we want to talk about how we only do this on the high-dimensional raw embeddings?**

Then, we also have the Davies-Bouldin index, where lower values indicate better clustering. We found that having 3, 6, or 20 clusters is ideal for the user embeddings projected into two dimensions (13). Due to the results from n _neighbors, we used the last layer embeddings to test cluster validity. Comparing these two results, we learn that dimension reduction aids us in finding clusters by cutting out dimensions with noisy information. Therefore, it is important to determine which features to keep when reducing the dimension rather than operating in the high dimensional space.

7 Next Steps

break up the "conclusion and next steps" section into two sections.

8 Conclusion

We were able to create a pipeline for producing and clustering user difference embeddings. Our baseline results for UMAP clustering includes that we were able to achieve distinct clusters. For the next step, we will try to create more meaningful clusters by experimenting with other models and layers to extract the emdbeddings from. Our baseline results with k-means clustering is a Davies-Boudin value of around 0.3. For our next iteration, we will be trying new clustering techniques to see if we can get a lower Davies-Boudin value. Lastly, we have recently obtained 5,000+ more utterances of data. Another future goal is to build an automated process that can handle this increased data volume.

9 Figures

I put all the images down here so that we can write the paper and then we can reformat them and put it back in later.

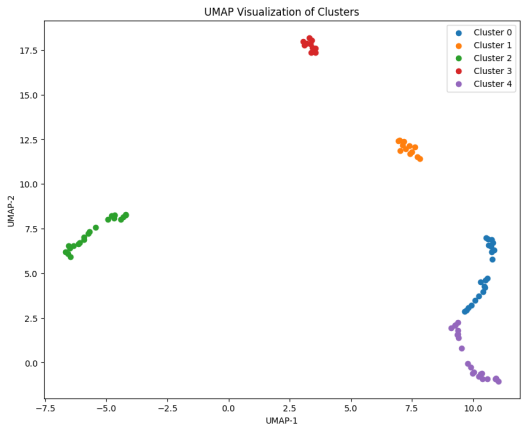

Figure 6: The metric parameter is set as cosine (all other parameters at default values). Embeddings were passed through the UMAP dimension reduction twice. Run on the second layer embeddings

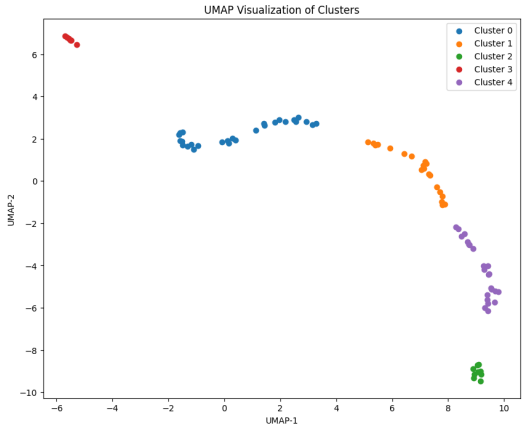

Figure 7: The metric parameter is set as cosine (all other parameters at default values). Embeddings were passed through the UMAP dimension reduction twice. Run on the eleventh layer embeddings

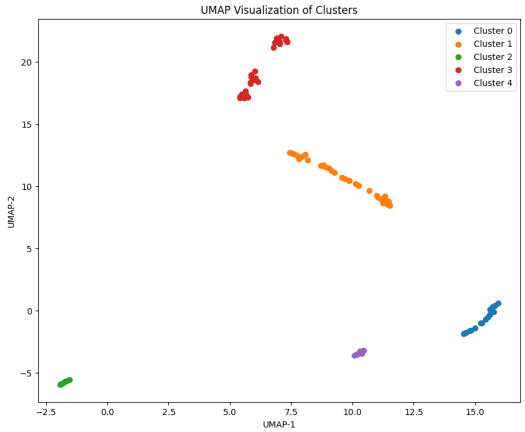

Figure 8: The metric parameter is set as cosine (all other parameters at default values). Embeddings were passed through the UMAP dimension reduction twice. Run on the last layer embeddings

		Evaluated on Reduced Embeddings	
Layer	**Clustering Configuration**	**Calinski-Harabasz**	**Davies-Bouldin**
2nd Layer	Baseline	76.596	0.857
2nd Layer	Decrease Num Neighbors	141.503	0.549
2nd Layer	Cosine Similarity	2903.195	0.481
2nd Layer	Num Neighbors and Cosine Similarity	**15201.540**	**0.052**
11th Layer	Baseline	79.529	0.912
11th Layer	Decrease Num Neighbors	215.737	0.428
11th Layer	Cosine Similarity	5647.906	0.409
11th Layer	Num Neighbors and Cosine Similarity	**13080.127**	**0.170**
Last Layer	Baseline	88.553	0.803
Last Layer	Decrease Num Neighbors	372.287	0.238
Last Layer	Cosine Similarity	2357.717	0.499
Last Layer	Num Neighbors and Cosine Similarity	**17178.887**	**0.045**

Table 1: Cluster Quality Metrics for Different Embedding Layers and Settings (Reduced Embeddings)

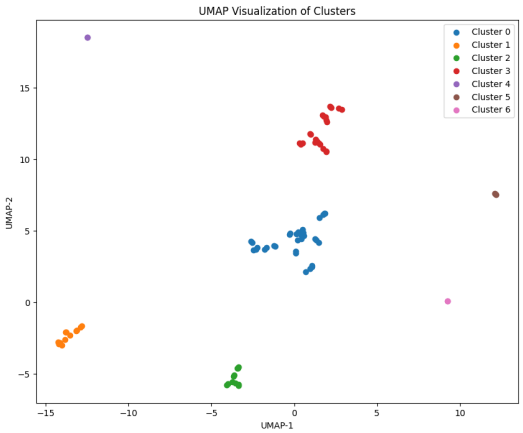

Figure 9: The n_neighbor parameter is set as 2 (all other parameters at default values). Run on the Last Layer Embeddings were passed through the UMAP dimension reduction once.

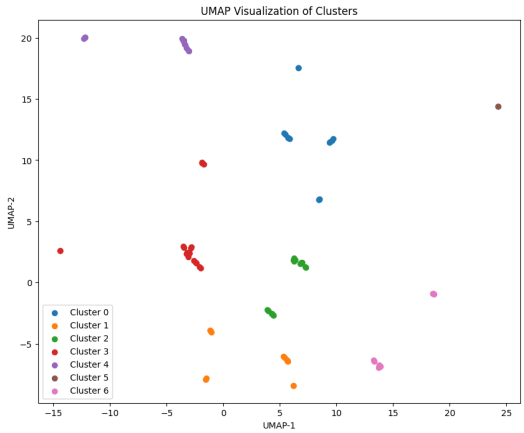

Figure 10: The n_neighbor parameter is set as 2 (all other parameters at default values). Run on the Second Layer Embeddings. Embeddings were passed through the UMAP dimension reduction once.

Layer	Clustering Configuration	Evaluated on Raw Embeddings	
		Calinski-Harabasz	**Davies-Bouldin**
2nd Layer	Baseline	**3.750**	2.866
2nd Layer	Decrease Num Neighbors	3.699	3.273
2nd Layer	Cosine Similarity	1.862	3.220
2nd Layer	Num Neighbors and Cosine Similarity	1.502	**2.388**
11th Layer	Baseline	3.603	2.937
11th Layer	Decrease Num Neighbors	**3.738**	3.688
11th Layer	Cosine Similarity	1.584	3.243
11th Layer	Num Neighbors and Cosine Similarity	1.305	**2.334**
Last Layer	Baseline	4.377	2.757
Last Layer	Decrease Num Neighbors	**7.456**	2.823
Last Layer	Cosine Similarity	2.118	3.168
Last Layer	Num Neighbors and Cosine Similarity	1.795	**2.329**

Table 2: Cluster Quality Metrics for Different Embedding Layers and Settings (Raw Embeddings)

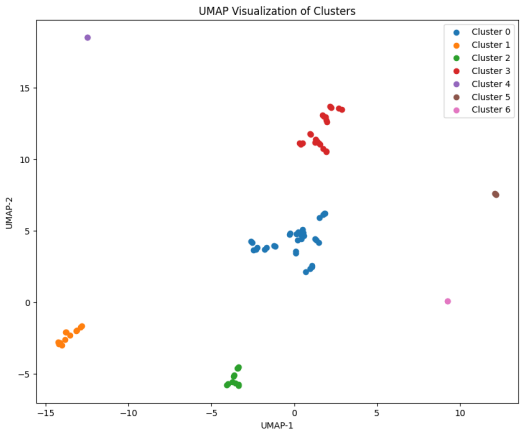

Figure 11: The n_neighbor parameter is set as 2 (all other parameters at default values). Run on the Eleventh Layer Embeddings. Embeddings were passed through the UMAP dimension reduction once.

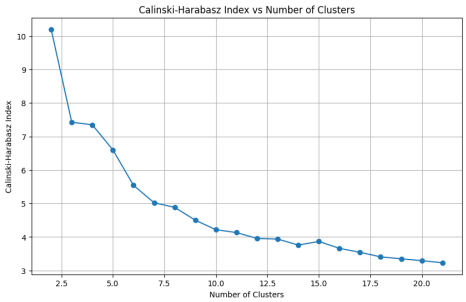

Figure 12: Calinski index as k-means clustering is applied on the raw high-dimensional user embeddings

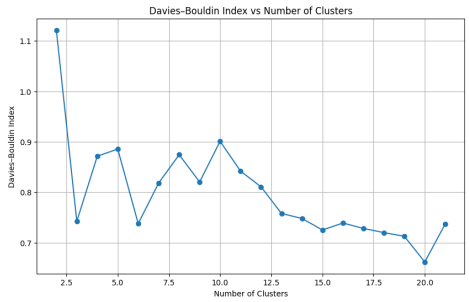

Figure 13: Davies-Bouldin index as k-means clustering is applied on user embeddings passed through UMAP

References

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. 2013. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Martijn Bartelds and Martijn Wieling. 2022. Quantifying language variation acoustically with few resources. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3735–3741, Seattle, United States. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Ankita Pasad, Bowen Shi, and Karen Livescu. 2023.

Comparative layer-wise analysis of self-supervised speech models.