INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Discovering Phonetic Feature Event Patterns in Transformer Embeddings

Patrick Cormac English[1], *John D. Kelleher*[2], *Julie Carson-Berndsen*[1]

SFI Centre for Research Training in Digitally-Enhanced Reality (d-real), Ireland
[1]ADAPT Research Centre, School of Computer Science, University College Dublin, Ireland
[2]ADAPT Research Centre, Technological University Dublin, Ireland

patrick.english@ucdconnect.ie, john.d.kelleher@tudublin.ie, julie.berndsen@ucd.ie

Abstract

Domain-informed probing of large speech recognition transformer-based models offers an opportunity to investigate how phonetic information is captured and transformed in the information-rich embeddings that emerge as part of the recognition process. Previous work in this area has established the efficacy of probing these embeddings with simple multi-layer perceptron models to identify the information patterns encoded at each layer of the transformer. This paper explores phonetic feature event patterns which evolve at each layer of a transformer model. Probing models are trained with phonetic embeddings, which are averaged and labelled at the phone level using the TIMIT dataset, to detect the presence of certain phonetic features in time-steps of a speech signal. This paper demonstrates how the detection of phonetic features within the embeddings of transformer models, such as voicing, frication and nasal, provides insights in relation to the encoding of speech patterns in these models.

Index Terms: phonetics, phonology, embeddings, transformers

1. Introduction

The development of large transformer models which are trained on large corpora of unlabelled data has seen significant improvement in standardised performance metrics such as word error rate (WER). These models generate internal vector representations (embeddings) of their inputs as part of their operation. There is a growing body of work exploring what types of information are encoded in the latent embeddings created by transformer models. One methodology for analysing these embeddings is known as *probing*. The intuition underlying probing is that if you can train a simple classifier to accurately predict from a transformer embedding of an input whether a feature was present in the input, then this provides evidence that the transformer model encodes information about this feature in its embeddings.

The internal processes of these transformer models remains an active area of investigation, with particular attention given to the remaining errors made by models of this architecture [1]. At present, full operation of an automatic speech recognition (ASR) transformer will typically output a token in a provided character set (graphemes, phonetic symbols, etc.) with errors only indicated through incorrect token predictions. However, while speech may be represented in terms of discrete segments, speech production is a continuous process, during which certain phonetic features may be present and overlap in time. [2].

The goal of this paper is to better understand how ASR transformer models process speech, with a particular focus on whether phonetic features (such as voicing, frication and nasal) are encoded in the embeddings generated by these models. Furthermore, given that transformer models typically include multiple layers of processing between the input and the output, we wish to analyse the evolution of the information encoded in these embeddings across the layers of the architectures. Identifying whether these features are present in the embeddings and, if they are, how the representations of these features emerge and evolve across the layers of these models may be useful, not only in terms of improving the performance of these models but also in providing a data-driven insight into how speech is recognised (e.g. which feature patterns emerge earlier from the data). Inspired by the probing methodology developed in natural language processing (NLP) to analyse text (word, sentence) embeddings, in this paper we adapt the probing methodology to apply it to speech embeddings. A key challenge in applying probing to speech is that speech is a continuous signal. Probing, however, assumes that the input is in discrete segments and that each input segment can be labelled as having the target features present or not. It is this labelling of segments that enables the training of the probes (the classifiers models) to predict from an embedding whether a target feature was present in the corresponding input segment. We use the information-rich embeddings of wav2vec 2.0 [3], which each represent a discrete time-step in an input signal, as our method of discretely representing speech.

The remainder of this paper is structured as follows: Section 2 outlines some recent work in this area that relates to our methodology. Section 3 describes the materials used followed by a description of the probing methodology in Section 4. Section 5 presents the results of the probing task and visualisations of some sample probe outputs. Section 6 concludes with some future work.

2. Related Work

2.1. Information Capture in Embeddings

In recent years there has been considerable interest in the practice of exploring neural embeddings of spoken language to identify the nature of information encoded within the embedding space of a given model. Such tasks typically require the introduction of a known supervision signal to allow for the relation of embeddings to the signal they represent. [4] used a synthetic language as the basis for evaluation, finding that contrastive elements identified in the embeddings examined could be related to articulatory features. Similarly, [5] used word2vec to create phoneme embeddings in order to investigate the extent to which similarity between the phonemes of English are influenced by distributional properties.

The embeddings generated by larger neural networks have also been explored on the basis that their complex architectures may be capable of more accurately encoding information

10.21437/Interspeech.2023-1985

found in an input signal than previous simpler models. [6] used an end-to-end ASR model to generate embeddings that were then explored to identify the categories of information captured. They noted that high-level aspects of the speech signal such as sound classes were identifiable in the embeddings generated, but reported much lower accuracy when considering more granular information such as phone label. [7] used a 3-layer model trained to differentiate between consonants and vowels. They conducted a domain-informed analysis of principal component analysis (PCA)-transformed embedding spaces, and their results were indicative of significant grouping based on vowel/consonant categorization and articulation style.

With the development of the transformer architecture [8], a high level of information capture has been identified in the speech embeddings of transformers trained on very large datasets of unlabelled data. Of particular relevance are the findings of [9], which probed non-averaged frames for a variety of features. They noted that evaluation performance fell as the granularity of phonetic knowledge required for distinction grew, with phone classification seeing the worst performance, but reported that high-level classification such as sonorance could be accurately identified within embeddings generated by a number of large transformer models.

2.2. Embedding Probes and Exploration

The probing methodology proposed by [10] has seen widespread use in exploring information capture within neural model embeddings. Probing involves the training of a classifier to identify the presence of an underlying feature (sentence properties such as length, parse tree depth etc.) within embedded representations of the signal. The performance of the probe with respect to classification accuracy is assumed to correlate with information capture in the embedding space. A number of works ([11, 12, 13, 14]) have made use of this methodology across a number of domains to associate embedded representations with feature occurrence to good effect. Similar work has been undertaken to identify layer-wise information capture in transformer embeddings, with the work of [15] identifying different levels of performance across layers with respect to feature identification.

[16] proposed an averaging method to identify wide capture of features within a number of large-transformer embedding spaces, including wav2vec2.0. As will be seen in Section 4.1, we adapt this averaging methodology as the basis for the data generation step in Section 4.1. Most recently, [17] proposed a method of knowledge exploration by which embedding spaces were explored for implicit knowledge capture within Large Language Models which found that binary contrastive inputs can be used effectively to explore the structure of the activations of a model, avoiding confounding factors in final model output.

3. Materials

3.1. TIMIT

The TIMIT read-speech corpus [18] was used for this task as it contains the timings for each of the phones in the set of utterances. This dataset contains 5.4 hours of 16 kHz spoken American-accented English audio in wav format. 8 US English dialects are represented in the dataset, with each speaker having 10 utterances. Each utterance is accompanied by a metadata file containing human-annotated phonetic and orthographic transcriptions and timings. The entire TIMIT dataset was used, with the TIMIT suggested training/test split.

3.2. wav2vec 2.0

The transformer architecture used for this task is the "base_960"[1] variant of the wav2vec 2.0 ASR model This model is comprised of the following components:

- a 1-D convolutional neural network, which takes a raw audio waveform W and outputs a latent speech representation, which is then processed by the transformer component.
- a 12-layer transformer module with an internal dimension of 768 and 8 attention heads which outputs contextual representation C.
- a language modelling head which divides output into a pre-selected vocabulary of 32 characters (English characters and a number of separators). The output of this layer is the transcription of the speech signal.

For the purposes of this paper, the embeddings of the transformer module for each of the 13 layers and the output of the CNN (layer 0), were examined; the language-modelling head was not used.

4. Experimental Methodology

4.1. Data Generation

Firstly we generated wav2vec 2.0 embeddings for each wav file in the TIMIT training and test datasets. This was done by running the wav2vec 2.0 model in its default configuration to get the embeddings for each transformer layer with the output_hidden_states=True parameter. As output, a [13*N*768] tensor was returned for each wav file (12 layers, with the additional layer 0 representing the output of the CNN), where N is the number of 25ms frames with 20ms stride and is dependent on the duration of the wav file. Each [1*768] representation is a *time-step representation*, corresponding to an slice of the speech signal This yielded the a dataset of time-step representations for each layer. In contrast to similar explorations of other large models [6], the representations for each layer are of the same dimensionality.

To associate each of the [1*768] time-step representations (one per 25ms of speech with 20ms stride, per layer - the wav2vec 2.0 model was pre-trained with these timings) with a phone label, the time-aligned annotations from the TIMIT dataset was used. A mapping was generated by comparing the relative position of each time-step representation in the [N*768] sequence, calculating the time-step represented by that embedding, and selecting the corresponding phone label for that point in time in the TIMIT annotation. At this point, the [258040*768] time-step samples derived from the TIMIT test set were set aside for the evaluation of the probes described in Section 4.2 below.

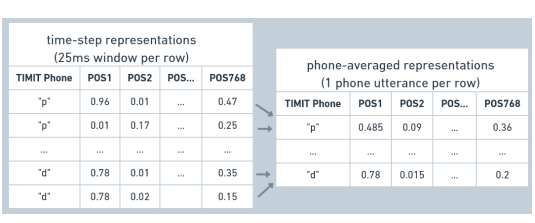

Figure 1: *Sample phone-averaging process*

[1] https://huggingface.co/facebook/wav2vec2-base-960h

For the samples generated from the TIMIT training set, using a modification of the method outlined in [16], the numeric values of all the embeddings occurring during a specific phone in the wav file were averaged to create *phone-averaged representations* and labelled with that phone as shown in Figure 1. This yielded 13 new datasets, one per layer, of 175232 phone-averaged representations.

Each phone-averaged representation in the 13 datasets was then annotated with feature labels based on a direct mapping from the phone labels, with a label added for each of the 3 chosen features (frication, nasal, voicing), to be used as input to the probing task. The 258040 time-step representations from the TIMIT test set were also labelled using the same method.

4.2. Probing Task

For the probing task, 39 MLP models (3 features, 13 layers) were trained to predict feature presence from the masked phone-averaged wav2vec 2.0 embeddings. The scikit-learn [19] implementation of the MLP was used, with 1 hidden layer of 200 ReLu activation neurons, and a single output neuron with logistic activation function. With the exception of the expanded hidden layer size of 200, all default scikit-learn hyperparameters were used.

Each MLP was trained on the phone-averaged representations of a given layer, and provided with the feature-presence label as a supervision signal. The training dataset for each model was the 175232 samples extracted from the TIMIT training dataset above. The 258040 samples generated from the TIMIT test dataset were reserved for model evaluation. Each model was trained on the [[175232]*768] representations of each layer, and provided with the [[175232]*1] feature presence labels (e.g. voicing - present/not present) as target category.

As the dimensionality of the time-step and phone-averaged representations are equivalent, it was then possible to use the probing models trained above to predict feature presence in time-steps based on patterns learned from the phone-averaged embeddings.

In order to ensure that probe performance was reflective of information relevant to the task, and not chance correlation, we created a separate set of randomly generated datasets to follow best practice [20]. These datasets were copies of the original training data, but with each feature column replaced by random values from within the feature column range. The 39 probes trained on these datasets were unable to effectively predict features in the training dataset, and were close to the random chance baseline on average.

4.2.1. Layer-Wise Feature-Detection Performance

Each model was evaluated with the [258040*768] test data, with feature-presence labels removed. Each probe was tested on the dataset from the layer that generated the training data for that probe. For each model, a set of [258040*1] feature-presence predictions were generated. From the above output, average accuracies per layer/feature were calculated.

4.2.2. Feature Detection in Time-Step Representations

In order to further investigate the feature detection of the MLP probes, we ran the probes on some sample utterances from the TIMIT dataset. These utterances were processed by the wav2vec 2.0 model in the same configuration as Section 4.1 to generate [13*N*768] outputs, where N is relative to the duration of the wav file. These are time-step representations.

For each layer in the output, the respective model/layer MLP was provided with the [N*768] representation of the time-step, returning an [N*1] array of binary feature presence predictions. These were then collated to generate an [Layer*N*1] matrix of layer-wise feature detections, which are discussed in Section 5.2. The outputs were then assessed to identify whether the time-step feature-presence predictions accorded with knowledge of feature presence in the respective time-steps.

5. Results and Discussion

5.1. Layer-Wise Feature-Detection Accuracies

Figure 2 presents the layer-wise accuracy scores for the voicing, frication and nasal features. The test samples were masked using the method described in Section 4.2.

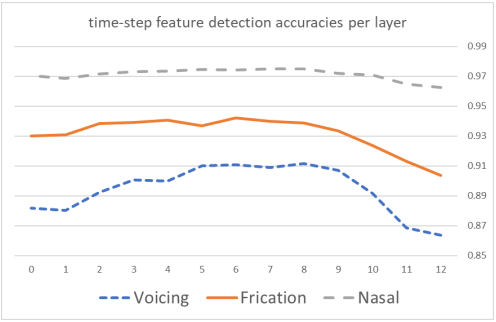

Figure 2: *Average feature label accuracies per layer*

Although accuracy is high overall, on both the voicing and nasal features the probe trained on the layer 8 embeddings achieved the best accuracy, whereas layer 6 is the highest-performing layer for frication detection. All feature-detection accuracies see some fall-off at layer 12. The decline seen in our results is potentially a result of the final layer embeddings being fine-tuned for the transcription task. This aligns with previous work e.g. [6] which noted an accuracy fall off in final layers in a similar task. An exploration of word embeddings within another transformer architecture [13] reported that the self-attention mechanism had resulted in a change in information encoded by the final layer to include significant amounts of contextual information in addition to token-specific information. We also note that there is some loss of information between the human-annotated phones in TIMIT and feature presence in a time-step, so these accuracies are only a partial reflection of model performance, which will be further examined with a sample utterance in the next section.

5.2. Feature Detection

For the purposes of illustration, a section of a sample TIMIT utterance, *"This is known as conformational entropy"*, was chosen to look more closely at feature detection at each probing layer. Specifically the section of the word ***conforma**tional*, which is highlighted in bold, contains the voiced, nasal and fricative phones.

5.2.1. Nasal, Voicing and Frication Features

Figure 3 displays the layer-wise detection of feature presence within the section of the word *conformational*. The top row indicates the TIMIT phone annotation for the respective time-step, which can be seen on the x-axis. Features are colour-

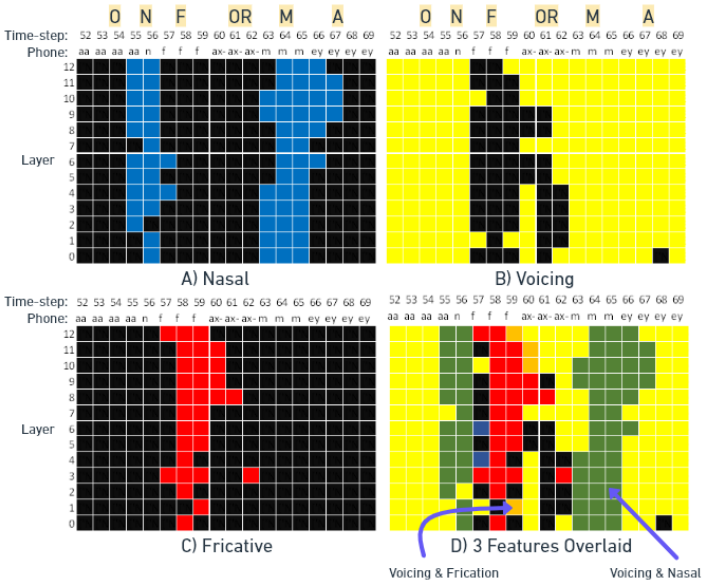

Figure 3: *Layer-wise detection of the 3 features in time-step representations of the section of the word "c**onforma**tional"*

coded based on detection in that time-step. Figure 3A depicts the layer-wise detection of nasal feature presence in blue. The presence of voicing can be seen in yellow in Figure 3B and the presence of frication is represented in red in Figure 3C. We note that the features tend to occur consistently within, and adjacent to, the time-steps where the human annotation indicated the presence of these features.

5.2.2. *Other Insights*

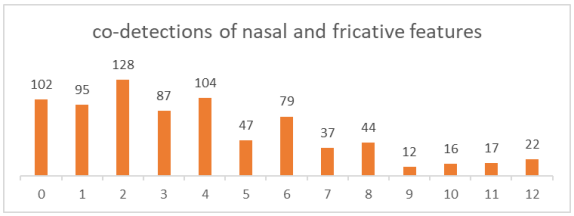

Figure 4: *Co-occurrence of nasal and frication features*

Figure 3D displays the detection of all three features across the same section of the word with green indicating the overlap of voicing and nasal and orange indicating the overlap of voicing and frication. Of interest here is the clear differentiation between the nasal and frication features. While, as expected, the voicing and nasal features, and the voicing and frication features do co-occur, the nasal and frication features do not overlap within a time-step. As the models predicting these features were trained independently, it is interesting that the probes appear to have identified that these features would be regarded as mutually exclusive as they are both manners of articulation. We explored this further by generating probe outputs for the 1680 wav files in the TIMIT test dataset. We observed that co-occurrence of the nasal and frication features was detected in 467 files. Figure 4 shows how many times per layer co-occurrence of nasal and frication was detected in these 467 utterances, with the least number, 12, occurring in layer 9. This layer is close to where the best accuracies were identified for the individual probes in Figure 2. The majority of these 12 co-occurrences were found to be cases where the nasal and fricative phones were in close proximity in the utterances.

This apparent constraint on co-occurrence of manner of articulation features is of particular interest for future work. Here the focus will be on a more comprehensive detection of all phonetic features to identify whether it is possible to automatically construct multi-tiered representations from the embeddings with a consistent feature geometry and where mutually exclusive features are represented on the same tier.

6. Conclusions and Future Work

This paper has demonstrated how the detection of phonetic features, such as voicing, frication and nasal, within the embeddings of transformer models, can provide insights in relation to the encoding of speech patterns in these models. We have demonstrated that phone-averaged representations may be used to train probes which are capable of identifying patterns of feature presence in time-step representations. We also found that the outputs of these probes, trained independently, appear to capture some constraints on feature co-occurrence. Future work will expand this approach to encompass a wider range of phonetic features with the goal of identifying whether the emergence of the feature patterns are coherent with theories such as multilinear phonology and feature geometry.

7. Acknowledgements

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106_P2) and is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

8. References

[1] Z. Zhang, Z. Wang, R. Kamma, S. Eswaran, and N. Sadagopan, "Patcorrect: Non-autoregressive phoneme-augmented transformer for asr error correction," 2023. [Online]. Available: https://arxiv.org/abs/2302.05040

[2] J. Carson-Berndsen, "Time map phonology: Finite state models and event logics in speech recognition," 1998.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Aul, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Neural Information Processing Systems (NeurIPS)*, 2020.

[4] S. Kolachina and L. Magyar, "What do phone embeddings learn about phonology?" *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2019.

[5] E. O'Neill and J. Carson-Berndsen, "The effect of phoneme distribution on perceptual similarity in English," *Proc. Interspeech 2019*, pp. 1941–1945, 2019.

[6] Y. Belinkov and J. R. Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," in *NIPS*, 2017.

[7] O. E. Scharenborg, N. van der Gouw, M. A. Larson, E. Marchiori, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, "The representation of speech in deep neural networks," *Lecture notes in computer science*, no. Part II, 2019.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[9] D. Ma, N. Ryant, and M. Liberman, "Probing acoustic representations for phonetic properties," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 311–315.

[10] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018.

[11] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019.

[12] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, "Linguistic knowledge and transferability of contextual representations," *CoRR*, vol. abs/1903.08855, 2019.

[13] V. Nedumpozhimana and J. Kelleher, "Finding bert's idiomatic key," in *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, 2021, pp. 57–62.

[14] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. R. Bowman, D. Das, and E. Pavlick, "What do you learn from context? probing for sentence structure in contextualized word representations," *CoRR*, vol. abs/1905.06316, 2019.

[15] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019, pp. 3651–3657.

[16] J. Shah, Y. K. Singla, C. Chen, and R. R. Shah, "What all do audio transformer models hear? probing acoustic representations for language delivery and its structure," *ArXiv*, vol. abs/2101.00387, 2021.

[17] C. Burns, H. Ye, D. Klein, and J. Steinhardt, "Discovering latent knowledge in language models without supervision," 2022. [Online]. Available: https://arxiv.org/abs/2212.03827

[18] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 11 1992.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[20] Y. Belinkov, "Probing classifiers: Promises, shortcomings, and advances," *Association for Computational Linguistics*, vol. 48, pp. 207–219, 2021.