

LAYER-WISE ANALYSIS OF A SELF-SUPERVISED SPEECH REPRESENTATION MODEL

Ankita Pasad, Ju-Chieh Chou, Karen Livescu

Toyota Technological Institute at Chicago

{ankitap, jcchou, klivescu}@ttic.edu

ABSTRACT

Recently proposed self-supervised learning approaches have been successful for pre-training speech representation models. The utility of these learned representations has been observed empirically, but not much has been studied about the type or extent of information encoded in the pre-trained representations themselves. Developing such insights can help understand the capabilities and limits of these models and enable the research community to more efficiently develop their usage for downstream applications. In this work, we begin to fill this gap by examining one recent and successful pre-trained model (wav2vec 2.0), via its intermediate representation vectors, using a suite of analysis tools. We use the metrics of canonical correlation, mutual information, and performance on simple downstream tasks with non-parametric probes, in order to (i) query for acoustic and linguistic information content, (ii) characterize the evolution of information across model layers, and (iii) understand how fine-tuning the model for automatic speech recognition (ASR) affects these observations. Our findings motivate modifying the fine-tuning protocol for ASR, which produces improved word error rates in a low-resource setting.

Index Terms— Self-supervised pre-training, representation analysis, speech representation learning

1. INTRODUCTION

Self-supervised learning (SSL) techniques leverage large-scale unlabeled data to learn meaningful representations [1–3]. In such techniques, the unlabeled data is used to design an input and corresponding target output, without any manual annotations. The learned representations are then used as input to a supervised model (and often fine-tuned) for a downstream task. The expected outcome is to either improve downstream task performance or to reduce the amount of labeled data required for training. For the speech domain, various SSL techniques have recently been shown to improve downstream task performance [4–13]. Although new and improved approaches are being proposed at a rapid rate, the pre-trained representations themselves are not well-understood, leaving the development and application of SSL models as a time- and resource-consuming process of trial and error.

We seek to fill this gap by analyzing pre-trained models

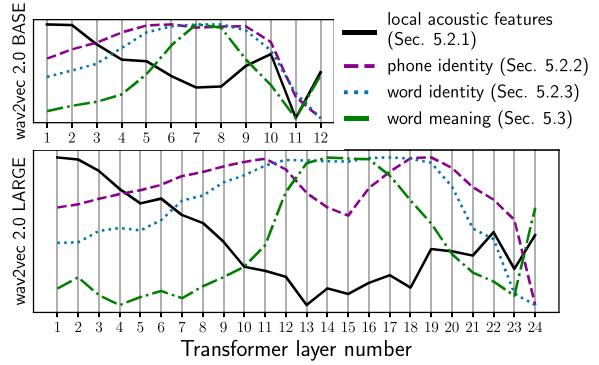


Fig. 1. Visualization of properties encoded at different W2V2 layers. The curves measure different metrics on different scales; they are shown together only to compare where major peaks and valleys occur. Details in the indicated sections.

to understand how the representations evolve across layers, how they relate to a range of linguistic properties, and how they change when fine-tuned for a downstream task. We are especially interested in developing tools to study representations directly, rather than training additional classifiers as probes, to avoid the overhead and unclear dependence on design decisions involved in training classifiers. In this work, we focus our analysis on the open-source wav2vec 2.0 (W2V2) models [11], which have been successful for speech recognition [14–16] and translation [17]. Our main findings are:

- The W2V2 transformer layers follow an autoencoder-style behavior, where as we go deeper into the model, the representation starts deviating from the input speech features followed by a reverse trend where even deeper layers become more similar to the input, as if reconstructing the input.
- The layer-wise evolution of the representations follows an acoustic-linguistic hierarchy, where the shallowest layers encode acoustic features, followed by phonetic, word identity, and word meaning information (and then followed by a reverse trend as described above), as illustrated in Fig. 1.
- Fine-tuning the model for ASR breaks the autoencoder-style behavior in the final few layers, which accordingly also get better at encoding word identity.
- The final convolutional (CNN) layers and initial transformer layers are highly correlated with mel spectrogram features, suggesting that the model learns to extract features similar to human-engineered ones.

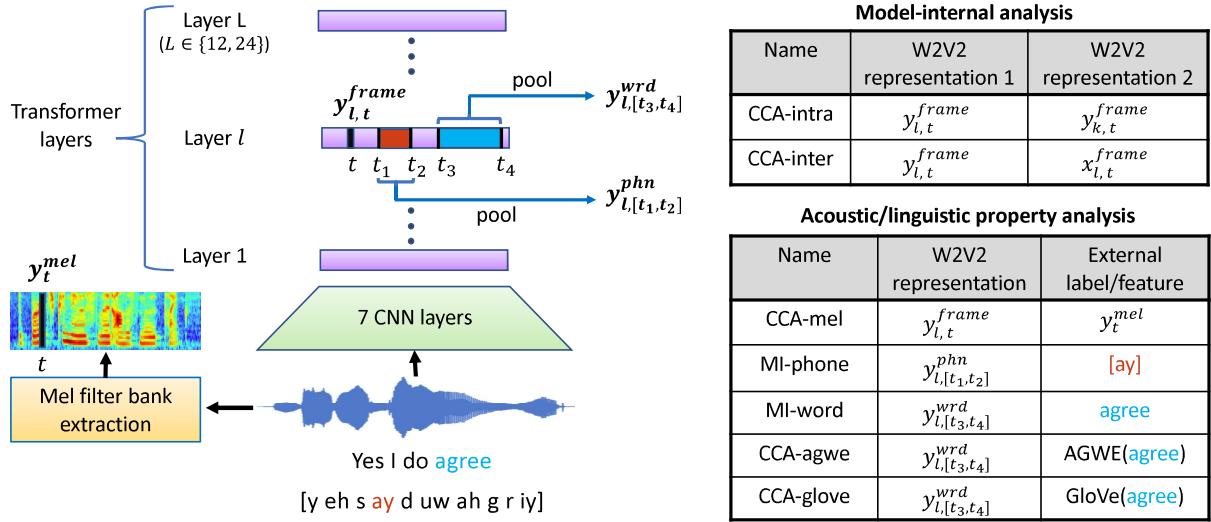


Fig. 2. Summary of our analyses using CCA and MI. Left: W2V2 architecture sketch. The Base and Large models have $L = 12$ and 24 transformer layers respectively. Right: Representations/labels used for each experiment. “Pool” refers to a pooling operation to combine frame representations into a phone/word segment representation, shown here for a segment corresponding to the phone [ay] and the word “agree”; in our experiments we use mean pooling. “AGWE” and “GloVe” refer to acoustically grounded word embeddings and GloVe word embeddings respectively. See Sec. 4.2 for details.

- The model encodes some word meaning information.
- The last two layers often defy the previous layers’ trends.
- A fine-tuning protocol, designed based on these findings, improves ASR performance in low-resource settings.

2. RELATED WORK

There has been extensive work on analyzing supervised speech models [18–20], but research on analyzing SSL models has been limited. Some very recent work has explored the phonetic, paralinguistic, and semantic content in SSL models using classifier probes [10, 16, 21, 22] and relationships between models with different training objectives and architectures [23]. The 2021 Zero Resource Speech Benchmark [24] introduces zero-shot analysis datasets and metrics to evaluate the ability of SSL speech representations to encode different levels of linguistic information. While we share much of the motivation of [21, 22, 24], we focus on layer-wise analysis of a range of acoustic-linguistic content using lightweight methods that don’t rely on training classifiers or collecting any additional labels for analysis, making it easier to scale.

Layer-wise analysis of linguistic structure has also been done before for visually grounded speech [25] and SSL text models [26]. Our methods of canonical correlation analysis (CCA) and discrete mutual information (MI) estimates are closest to Voita *et al.*’s work on text models [27]. MI has also been used to analyze supervised ASR models [20]. Unlike prior work, we apply these methods to the analysis of the relationship between representations and both discrete labels and continuous embeddings, and between representations from pre-trained and fine-tuned models. To our knowledge, this is the first work to analyze an SSL speech model on a range of linguistic properties using non-parametric probes.

3. ANALYSIS METHODS

Fig. 2 sketches the W2V2 model structure and the representations used in many of our analyses.

Canonical Correlation Analysis. CCA [28] is a statistical technique that measures the relationship between two continuous-valued random vectors as represented by the maximum correlations between their linear projections. CCA has been previously used as a measure of similarity to compare representations within and across neural network models [27, 29, 30]. Here we use it in the same way, and also to measure the similarity between a layer representation and another vector, such as word embeddings or acoustic features.

CCA takes n pairs of vectors $\{(x_1, y_1), \dots, (x_n, y_n)\}$, sampled from the random vectors (or “views”) $X \in \mathbb{R}^{d_1}, Y \in \mathbb{R}^{d_2}$, as input and returns a correlation score as a measure of similarity between the two views. The solution can be defined iteratively as follows: First we define the directions of maximum correlation between linear projections of X and Y : $v_1, w_1 = \arg \max_{v,w} \text{corr}(v^T X, w^T Y)$. The subsequent directions $v_i, w_i \forall i \in [2, k], k = \min(d_1, d_2)$, maximize the same correlation subject to each new projection being uncorrelated with others in the same view.

In standard CCA the canonical correlation $\text{CCA}(X, Y)$ is the sum (or mean) of the correlations $\rho_i = \text{corr}(v_i^T X, w_i^T Y)$. We use a variant, *projection-weighted CCA* (PWCCA) [31], which computes a weighted mean of the ρ_i s, with higher weights for directions accounting for a higher proportion of the input. PWCCA has been found to be more robust to spurious correlations in the data. Since PWCCA is asymmetric, we report the mean of the two quantities $\text{CCA}(X, Y)$ and $\text{CCA}(Y, X)$. Henceforth we refer to this average as the

“CCA similarity”, and it has a maximum value of 1.

As illustrated in Fig. 2, we use PWCCA to measure similarity between the W2V2 layer representations and various continuous-valued quantities of interest, either (i) from a different layer of the same model (*CCA-intra*), (ii) from a fine-tuned version of the model (*CCA-inter*), or (iii) from an external representation. For the third type of analysis we use mel filter bank features (*CCA-mel*), acoustically grounded word embeddings [32] (*cca-agwe*)¹ and GloVe word embeddings [33] (*cca-glove*) as ways to assess the local acoustic, word-level acoustic-phonetic, and word meaning information encoded in the W2V2 representations respectively.

Mutual information. While CCA is natural for relating continuous-valued vectors, we use mutual information (MI) to measure dependence between the representations, y^{phn} or y^{wrd} from Fig. 2, and the corresponding phone (*MI-phone*) or word (*MI-word*) label. We cluster the continuous-valued representations to obtain discrete clusters, as in [20, 27].

Word discrimination. (*word-disc*) is the task of detecting whether two speech segments correspond to the same or different words [34] and is commonly used to evaluate acoustic word embeddings and other acoustic representations [35–38]. We follow a typical evaluation protocol, where we label a pair of segments as “same word” if the cosine similarity between their word-level representations is above some threshold, and measure performance via the average precision as the threshold is varied. We use this task-specific measure primarily to corroborate our findings from *MI-word*.

Word similarity tasks. We perform a suite of 11 standard word similarity tasks (*word-sim*) [39] as an additional measure of word meaning information.² We extract context-independent word embeddings from W2V2, as described in Sec. 4.2. The semantic similarity score for each word pair is measured as the cosine similarity between these embeddings. Performance is measured as the Spearman’s ρ correlation between these scores and the human similarity judgements.

4. EXPERIMENTAL SETUP

4.1. Representation Learning Model

The W2V2 model [11] maps raw waveforms to higher-level contextual features via a set of convolutional layers followed by self-attention (transformer) layers, as shown in Fig. 2, and is trained with a contrastive objective that measures the ability of the model to differentiate between a true masked input segment and a set of distractors. The self-attention layers in the transformer allow the model to encode information from the context surrounding a given masked segment.

We analyze three W2V2 variants: (i) *Base*: 12 layers, trained on 960 hours of LibriSpeech [40], (ii) *Large-960*: 24 layers, trained on 960 hours of LibriSpeech, (iii) *Large-60k*:

¹AGWEs are trained to be close to acoustic embeddings of the corresponding words, so we expect they encode mainly acoustic-phonetics.

²<https://github.com/vecto-ai/word-benchmarks>

Experiment	# labels	# representation examples
CCA-intra, CCA-inter, CCA-mel	n/a	150k frames
CCA-agwe, CCA-glove	2.7k words	4.8k word segments
MI-phone	39 phones	train: 187k phone segments dev: 7.6k phone segments
MI-word	500 words	train: 427k word segments dev: 6.9k word segments
word-disc	300 words	2.4k word segments (2.9M pairs)

Table 1. Data subsets curated for our analysis. We repeat each experiment on four sample sets. The numbers here are averages across the four sets. For MI experiments, the train subsets are used to define the clustering. For word-disc, we use words that are at least 5 characters and 500ms long.

24 layers, trained on 60k hours of LibriVox [41]. We also analyze W2V2 fine-tuned for ASR with 10 minutes (*ft-10m*), 100 hours (*ft-100h*), and 960 hours (*ft-960h*) of labeled LibriSpeech.³ Fine-tuning consists of adding a randomly initialized linear layer to the pre-trained model, and then training with character-level connectionist temporal classification (CTC) loss [42], while keeping the CNN layers frozen [11]. We refer to this as the “standard approach” in Sec. 6.

4.2. Setup Details

We perform all experiments on LibriSpeech. The sampled utterances (details in Tab. 1) are passed through each W2V2 model, and the outputs from all layers are extracted. Random masking is turned off except for experiments analyzing the effect of masking (Sec. 5.5).

Representation extraction: We use LibriSpeech alignments generated using the Montreal forced aligner [43, 44] to define phone and word segments. As illustrated in Fig. 2, word-level representations y^{wrd} are obtained by averaging the frame representations of all frames in a given word segment. Phone-level representations y^{phn} are obtained by averaging the frame representations of the central third of each phone segment; the first and last third are discarded to reduce co-articulation effects. These segment representations are used for all experiments in Tab. 1 except the first row. The context-independent embedding (used for the *word-sim* experiments) for each word is computed by averaging the y^{wrd} representations across all the instances of that word in train-clean.⁴

Mel filter bank features: 80-dimensional mel filter bank (fbank) features are extracted using a frame length of 25ms and an overlap of 10ms. In order to make the W2V2 representations comparable to the fbank features, we compute moving averages of CNN features or downsample fbank features as needed to ensure their strides and receptive fields match.

³All models are downloaded from the wav2vec 2.0 repository: <https://github.com/pytorch/fairseq/blob/master/examples/wav2vec>

⁴We also tried weighted mean pooling, using averaged attention weights from all the attention heads, which produced similar results.

Discrete cluster IDs: For MI experiments, we discretize the continuous-valued W2V2 representations. Specifically, we cluster a set of phone/word-level representations sampled from the train-clean LibriSpeech split with roughly the same number of examples of each label,⁵ using mini-batch k-means with k=500 for MI-phone and k=5000 for MI-word, and assign each development set example to the nearest cluster.

5. FINDINGS

We present results for experiments done on the dev-clean split on some of the W2V2 variants (base, base-ft-960, large-60k, large-60k-ft-960); the findings generalize to dev-other and to the large-960 model unless stated otherwise. We analyze pre-trained models in Sec. 5.1-5.3 and their fine-tuned counterparts in Sec. 5.4.⁶ Each plot below gives the mean of the relevant measure across the four sample sets; typical variation across the sets is < 0.02 for CCA measures, < 0.07 for MI, and $< 2\%$ for word-disc. We refer to the output of transformer layer l as the representation at layer l and the output of the CNN feature encoder as layer 0 or “local features”.

5.1. How do the representations evolve across layers?

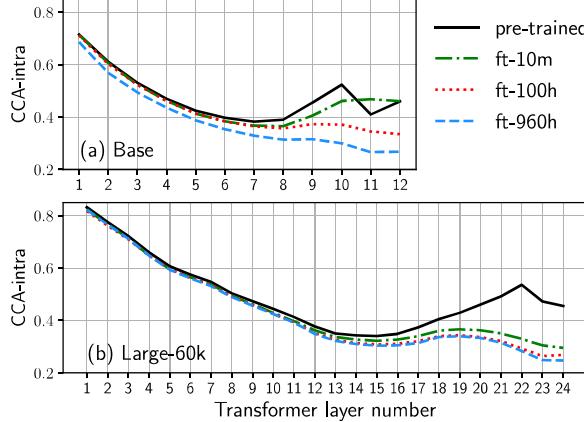


Fig. 3. CCA similarity with local features.

In Fig. 3 we compare (via CCA similarity) the transformer layer representations with the “local features” extracted by the CNN module (layer 0). We see that the pre-trained model (solid black curve) follows an autoencoder-style behavior, where as we go deeper into the model, the representation starts deviating from the input features, followed by a reverse trend where even deeper layers become more similar to the input, as if reconstructing the input (although this trend seems to break for the last two layers; see Sec. 5.5). Since the training objective is to distinguish the masked input segment from distractors, it is natural for the final layers to have similar properties to the input. A similar behavior, referred to as context-encoding and reconstruction, has been observed for the BERT text model [27], where the objective is based on masked reconstruction rather than contrastive prediction.

⁵Similar trends are obtained when the chosen instances are uniformly sampled from the data instead.

⁶The Sec. 5.1-5.3 figures combine both pre-trained and fine-tuned models.

5.2. Where is acoustic/linguistic information encoded?

Next we consider how certain properties are encoded in different layers. As a reminder, all our experiments are performed on features extracted locally from a short span of frames (frame/phone/word-level). Any increase in “information” across layers for these local representations is possible due to contextualization from the self-attention layers that enable each frame-level output to access the whole utterance. For the same reason, any decrease in “information” across layers could be attributed to de-localization, i.e. the information is no longer localized to the frame/phone/word segment.

5.2.1. Frame-level acoustic content

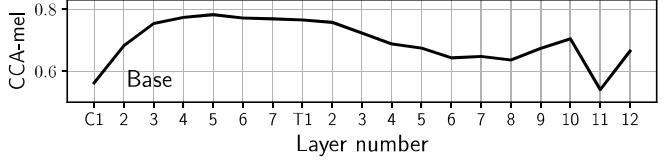


Fig. 4. CCA similarity between layer representations and fbank; Ci : CNN layer i ; Tj : transformer layer j .

Fig. 4 shows the layer-wise CCA similarity between fbank vectors and Base model layers. For the first few layers the correlation increases with depth. The Large models follow a similar curve, with high correlation for layers $C4-T2$ (> 0.75). We can infer that the model learns to compute features much like fbank, suggesting a potential simplification to W2V2 to take fbank as input (which we leave to future work).

5.2.2. Phonetic information

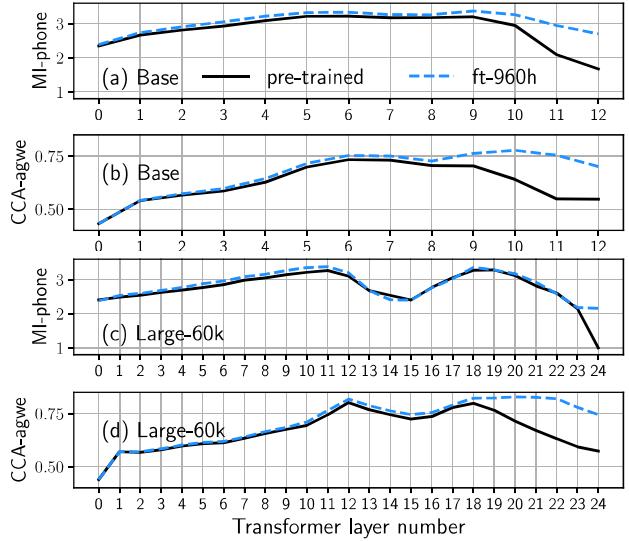


Fig. 5. MI with phone labels (max: 3.6) and CCA similarity with AGWE.

We measure the phonetic information encoded in the pre-trained model in two ways, MI-phone and CCA-agwe⁷ (see

⁷We use AGWEs trained on LibriSpeech similarly to [32].

Sec. 3), both shown in Fig. 5. We expect AGWEs to encode mostly phonetic information, and indeed the phone and AGWE curves in Fig. 5 follow broadly similar trends.

We notice that phonetic information appears to be most salient around layer 6-7 for Base (similarly to [10], which includes a similar experiment). For Large-60k (Fig. 5), however, layers 11 and 18/19 appear equally adept at encoding phonetic information, with a drop in between.

5.2.3. Word identity

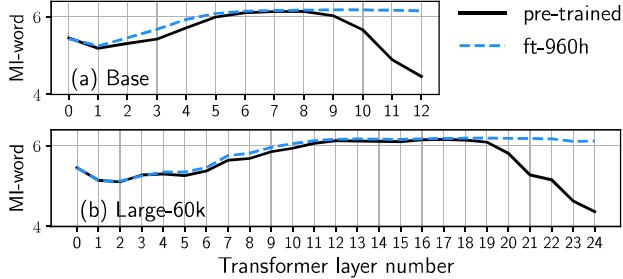


Fig. 6. MI with word labels (max: 6.2).

Fig. 6 shows the mutual information between the layer representations and word labels. For Base, the trends are similar to those of MI-phone (Fig. 5a). For Large-60k (Fig. 6b), word identity appears to be encoded similarly well by layers 12 to 18, without the dip seen in the MI-phone curve.

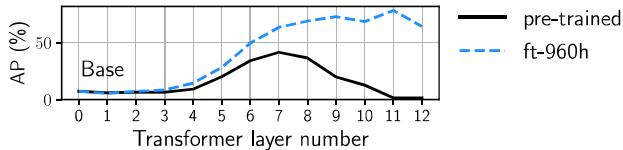


Fig. 7. Average precision (AP) for word discrimination.

As another measure of word identity content, Fig. 7 shows word discrimination performance, which follows a similar trend to MI-word (Fig. 6a). We experiment with all of the W2V2 variants (including ones not shown here), and find that MI-word and word-disc are always highly correlated.

5.2.4. Evidence for the most contextual layers

For the Large-60k model, the curves measuring acoustic-phonetic information (Figs. 3b, 5c, 5d) all have a dip around layers 13-17 (see also Fig. 1). These are also the same layers that seem to have the *most* word content (Fig. 6b). This suggests that around these layers, the model may be extracting the most contextual and high-level information, and retaining less lower-level information like phonetic content. Beyond these layers, the model enters the reconstruction phase, thus encoding more local representations at even deeper layers.

The Base model does not have the same significant intermediate drop for phonetic content (Figs. 1, 5a, 5b) as does Large-60k, which could indicate less contextualization. In experiments on Large-960 (not shown here), the MI-phone and CCA-agwe scores do not show this drop either, implying that this effect is the result of the larger training set of Large-60k, and not its larger model size.

5.3. Does the pre-trained model learn word meaning?

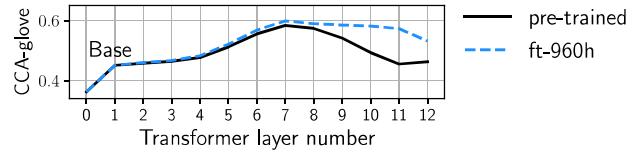


Fig. 8. CCA similarity with GloVe embeddings.

While some linguistic properties seem essential for the model to learn to solve the self-supervised task, it is not obvious that word meaning is one such property. We probe for word meaning in W2V2 by measuring the CCA similarity between word segment representations and the popular text-based GloVe embeddings [33], shown in Fig. 8. These plots (also a part of Fig. 1) provide further evidence that the central layers (7-8 for Base and 14-16 for Large-60k) encode the most contextual information. Note that these curves have a narrower plateau of maximum performance around these layers than the MI-word curves (Fig. 6), suggesting that the most contextual layers are better at encoding word meaning while the peripheral layers are good at encoding lower-level linguistic content but not meaning.

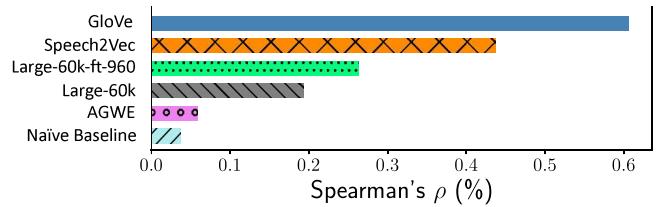


Fig. 9. Word similarity performance (mean across tasks).

To further calibrate our measure of semantic information, we evaluate the W2V2 representations on standard word similarity benchmarks, as described in Sec. 3. Fig. 9 reports the performance of the best layers for Large-60k and Large-60k-ft-960. The best performance for both models occurs at layer 15, which again agrees with our hypothesis that layers 14-16 contain the most semantic information.

We also present two baselines: (i) the *naive baseline* defines word distance as the character edit distance for each word pair; this baseline has non-trivial performance when orthography is a helpful clue (ii) the *AGWE baseline* uses AGWEs in place of the W2V2 representations, and may succeed for word pairs where acoustic-phonetic similarity correlates with semantic similarity. We also include two models that are trained specifically to encode semantics: (i) Speech2Vec [45] learns word embeddings from speech using an approach similar to word2vec [46] and is trained on LibriSpeech, and (ii) GloVe embeddings [33]. Since W2V2 is not trained with an explicit semantic criterion, it is not surprising that it is outperformed by Speech2Vec and GloVe. It is interesting, however, that W2V2 representations perform better than the non-semantic baselines, suggesting that some meaning is being encoded.

5.4. How does fine-tuning affect the above observations?

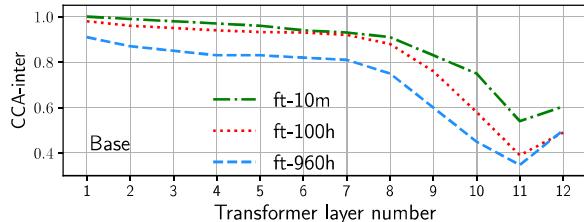


Fig. 10. CCA similarity between each layer of a pre-trained model and the same layer of fine-tuned models.

We see in CCA-intra, Fig. 3, that fine-tuning breaks the autoencoder-style behavior. After fine-tuning for ASR, the deeper layers that were originally reconstructing the input are now diverging from the input, and presumably learning more task-specific information. We also see from Fig. 10 that the higher layers change the most in fine-tuning, suggesting that the pre-trained model may not serve as a good initialization of these top layers for ASR. This finding suggests re-initializing these layers before fine-tuning, as has been recently discovered for BERT [47]. We design a fine-tuning experiment to validate this idea, described in Sec. 6.

MI-word consistently improves across the top layers (19-24) after fine-tuning (Fig. 6). The same does not always hold for phone identity (Fig. 5). These results indicate that, as might be expected, fine-tuning with character-level CTC loss is more directly related to the word identity than to phone identity. For the semantic measures (CCA-glove and word-sim) as well we don't see the same large improvements as for MI-word, again as may be expected since ASR does not necessarily require high-quality word meaning representations.

5.5. What about those peculiar last two layers?

We see a peculiar pattern in most of the CCA similarity curves for pre-trained W2V2 models, where at least one of the last two layers fails to follow the trend of the previous layers. We find that this peculiarity disappears when we turn random masking on and consider only the representations of the masked segments. Moreover, the phonetic and word content, as measured by MI, improves for the last two layers (while reducing for the rest) when working with the representations of masked segments. This finding suggests that the representations of the final two layers are more meaningful when the input segment is masked. Furthermore, this discrepancy is not present in the fine-tuned models, suggesting that this effect is connected to the training objective, but the exact relationship is unclear. We also note that this peculiarity has been observed for local representations extracted from BERT [48].

6. PRACTICAL IMPLICATIONS FOR ASR

We have noted that the last few layers of W2V2 change the most during fine-tuning (Fig. 10), and that the linguistic content that should be helpful for ASR is less well represented in the final few layers (Figs. 5a, 6a). Based on these observations, we hypothesize that some of these final layers do not

train set	n	standard → re-init 12-n layers	
		test-clean	test-other
10m	9	49.0 → 44.1	56.7 → 51.8
1h	11	20.3 → 19.8	29.8 → 29.3
10h	11	11.3 → 10.9	20.6 → 19.4

Table 2. Word error rates (%) for the modified fine-tuning protocol for the Base model, using the best value of n based on dev-clean performance, compared to standard fine-tuning. $A \rightarrow B$ indicates that standard fine-tuning produces WER A, and the proposed protocol produces WER B.

provide a good initialization for the task. To test this hypothesis we modify the “standard approach” by re-initializing the top layer(s) before fine-tuning. We conduct all ASR experiments using the SpeechBrain toolkit [49]. We experiment with W2V2-base and find that re-initializing the final 1-3 layers indeed outperforms the standard approach of initializing all layers from the pre-trained model (Tab. 2), with large improvements when fine-tuning on the 10-minute training set and minor improvements for larger training sets.

7. CONCLUSION

We have presented a set of analyses to assess the layer-specific information in pre-trained speech representations, applied to wav2vec 2.0 models. We find that various acoustic and linguistic properties tend to be encoded in different layers, and the pre-trained model follows an autoencoder-style behavior. We also find that the model encodes some non-trivial word meaning information, although more work is needed to determine the nature of the semantic content. We corroborate most of our findings with multiple analytical measures and certain downstream tasks. Such analyses can help understand the abilities and limitations of models trained without external supervision, and also help direct research toward additional useful modifications. For example, some of our findings have motivated a modification to the fine-tuning protocol, which leads to improved downstream ASR performance in the very low-resource setting.

Our analyses focus on representations extracted locally (over a frame/phone/word), so it does not measure the information delocalization that may be happening as a result of the self-attention layers. We leave in-depth analysis of self-attention to future work. Additional future directions include applying the same analytical tools to additional models with different architectures or training objectives, and further studying the implications for additional downstream tasks.

Acknowledgements. We thank Shane Settle for providing the AGWEs, and David Yunis, Puyuan Peng, and Shubham Toshniwal for help with preliminary experiments and ideation. This research was funded by NSF award IIS-1816627, by Air Force Office of Scientific Research award FA9550-18-1-0166, and by an AWS Machine Learning Research Award.

8. REFERENCES

- [1] Carl Doersch, Abhinav Gupta, and Alexei A Efros, “Unsupervised visual representation learning by context prediction,” in *CVPR*, 2015.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.
- [3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, “Improving language understanding by generative pre-training,” *Technical Report, OpenAI*, 2018.
- [4] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [5] Santiago Pascual, Mirco Ravanelli, Joan Serra, Antonio Bonafonte, and Yoshua Bengio, “Learning problem-agnostic speech representations from multiple self-supervised tasks,” in *Interspeech*, 2019.
- [6] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in *Interspeech*, 2019.
- [7] Yu-An Chung and James Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *ICASSP*, 2020.
- [8] Weiran Wang, Qingming Tang, and Karen Livescu, “Unsupervised pre-training of bidirectional speech encoders via masked reconstruction,” in *ICASSP*, 2020.
- [9] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in *ICASSP*, 2020.
- [10] Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “HuBERT: How much can a bad teacher benefit ASR pre-training?,” in *ICASSP*, 2021.
- [11] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *NeurIPS*, 2020.
- [12] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Kathrin Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *ICASSP*, 2020.
- [13] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., “SUPERB: Speech processing universal PERformance benchmark,” in *Interspeech*, 2021.
- [14] Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, et al., “Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training,” *arXiv preprint arXiv:2104.01027*, 2021.
- [15] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli, “Unsupervised cross-lingual representation learning for speech recognition,” *arXiv preprint arXiv:2006.13979*, 2020.
- [16] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, “Unsupervised speech recognition,” *arXiv preprint arXiv:2105.11084*, 2021.
- [17] Changhan Wang, Anne Wu, Juan Pino, Alexei Baevski, Michael Auli, and Alexis Conneau, “Large-scale self- and semi-supervised learning for speech translation,” *arXiv preprint arXiv:2104.06678*, 2021.
- [18] Yonatan Belinkov and James Glass, “Analysis methods in neural language processing: A survey,” *TACL*, vol. 7, pp. 49–72, 2019.
- [19] Shruti Palaskar, Vikas Raunak, and Florian Metze, “Learned in speech recognition: Contextual acoustic word embeddings,” in *ICASSP*, 2019.
- [20] Archiki Prasad and Preethi Jyothi, “How accents confound: Probing for accent information in end-to-end speech recognition systems,” in *ACL*, 2020.
- [21] Danni Ma, Neville Ryant, and Mark Liberman, “Probing acoustic representations for phonetic properties,” in *ICASSP*, 2021.
- [22] Jui Shah, Yaman Kumar Singla, Changyou Chen, and Rajiv Ratn Shah, “What all do audio transformer models hear? Probing acoustic representations for language delivery and its structure,” *arXiv preprint arXiv:2101.00387*, 2021.
- [23] Yu-An Chung, Yonatan Belinkov, and James Glass, “Similarity analysis of self-supervised speech representations,” in *ICASSP*, 2021.
- [24] Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux, “The Zero Resource Speech Benchmark 2021: Metrics and baselines

for unsupervised spoken language modeling,” in *Self-Supervised Learning for Speech and Audio Processing Workshop @ NeurIPS*, 2020.

[25] Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi, “Representations of language in a model of visually grounded speech signal,” in *ACL*, 2017.

[26] Ian Tenney, Dipanjan Das, and Ellie Pavlick, “BERT rediscovers the classical NLP pipeline,” in *ACL*, 2019.

[27] Elena Voita, Rico Sennrich, and Ivan Titov, “The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives,” in *NAACL*, 2019.

[28] Harold Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[29] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein, “SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability,” in *NIPS*, 2017.

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton, “Similarity of neural network representations revisited,” in *ICML*, 2019.

[31] Ari S Morcos, Maithra Raghu, and Samy Bengio, “Insights on representational similarity in neural networks with canonical correlation,” in *NeurIPS*, 2018.

[32] Shane Settle, Kartik Audhkhasi, Karen Livescu, and Michael Picheny, “Acoustically grounded word embeddings for improved acoustics-to-word speech recognition,” in *ICASSP*, 2019.

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning, “GloVe: Global vectors for word representation,” in *EMNLP*, 2014.

[34] Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, “Rapid evaluation of speech representations for spoken term discovery,” in *Interspeech*, 2011.

[35] Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater, “Unsupervised neural network based feature extraction using weak top-down constraints,” in *ICASSP*, 2015.

[36] Yushi Hu, Shane Settle, and Karen Livescu, “Multilingual jointly trained acoustic and written word embeddings,” in *Interspeech*, 2020.

[37] Robin Algayres, Mohamed Zaiem, Benoît Sagot, and Emmanuel Dupoux, “Evaluating the reliability of acoustic speech embeddings,” in *Interspeech*, 2020.

[38] Christiaan Jacobs, Yevgen Matusevych, and Herman Kamper, “Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation,” in *SLT*, 2021.

[39] Manaal Faruqui and Chris Dyer, “Community evaluation and exchange of word vectors at wordvectors. org,” in *ACL: System Demonstrations*, 2014.

[40] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “LibriSpeech: An ASR corpus based on public domain audio books,” in *ICASSP*, 2015.

[41] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., “Libri-light: A benchmark for ASR with limited or no supervision,” in *ICASSP*, 2020.

[42] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, 2006.

[43] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi,” in *Interspeech*, 2017.

[44] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, “Speech model pre-training for end-to-end spoken language understanding,” in *Interspeech*, 2019.

[45] Yu-An Chung and James Glass, “Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech,” in *Interspeech*, 2018.

[46] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013.

[47] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi, “Revisiting few-sample BERT fine-tuning,” in *ICLR*, 2020.

[48] Kawin Ethayarajh, “How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings,” in *EMNLP*, 2019.

[49] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al., “SpeechBrain: A general-purpose speech toolkit,” *arXiv preprint arXiv:2106.04624*, 2021.