SpeechBrain Scaling Study

Nishka Katoch, Artem Ploujnikov
DIRO
Université de Montréal
Montréal, QC
nishka.katoch@umontreal.ca artem.ploujnikov@umontreal.ca

Abstract

The present work aims to study the scaling properties of speech recognition and natural language understanding (NLU) models that take advantage of representations acquired via unsupervised learning in multispeaker settings. This work examines the scaling of a CTC-based speech recognition model with a pretrained wav2vec2 encoder, as well as a bimodal Natural Language Understanding (NLU) intent classification model derived from Tie Your Embeddings Down by Agrawal et al, which embeds text and audio commands in a shared embedding space with a shared classifier with different approaches to audio preprocessing, including raw features as originally proposed, the latent features from a RAVE variational autoencoder and a wav2vec2 acoustic model. We empirically derive scaling laws for the models tested: a power law for ASR and audio accuracy and a logarithmic law for text-to-audio embedding cosine similarity. We also perform a qualitative exploration of the shared text-audio latent space for the bimodal NLU model.

1 Introduction

Speech recognition and synthesis is a well-studied problem in artificial intelligence. Modern applications of speech processing are ubiquitous in consumer-grade personal assistants integrated with smart home devices, automated captioning, vehicle accessory control, accessibility devices, automated voice translation and automated voice response systems. Current research aims at both improving the fundamental properties and performance of such speech systems (speech recognition accuracy, the naturalness of speech synthesis, support for languages, emotions) and the integration of speech into the context of multimodal sensory processing integrating auditory and visual stimuli and conversation context.

Raw speech data consists of waveforms, which can be thought of as long single-dimensional tensors of audio samples with typical sampling rates ranging from 8Hz to 16KHz. Conventional neural approaches to speech processing involve an initial fixed transformation to a more condensed 2D representation based on a Fourier transform, such as a spectrogram (linear or MEL) or mel-frequency cepstral coefficients (MFCC)(1). While such representation are successful at condensing sound data to a format more easily processed by neural models (e.g. sequence-to-sequence(2) or Transformer(3)). In recent years, pretrained acoustic models of speech have been increasingly used for such tasks, offering significant advantages over fixed transformations, particularly in smaller-data regimes.

The main goal of the present work is to study the scaling and transfer properties of various neural representations of human speech in a wider context of making progress towards the development of universal representations of human speech that may be used for a variety of downstream tasks including speech recognition, speech synthesis, speaker identification and end-to-end translation in both unimodal and multimodal settings. Our work focuses on the use of such models for Automatic Speech Recognition and Natural Language Understanding.

Preprint. Under review.

We perform the experiments within the SpeechBrain(4) toolkit and enhance it with a new implementation of the NLU(5) common latent space model.

2 Related Work

Speech processing is dependent on using a suitable representation of speech inputs that can be fed to a deep learning model. Raw speech audio data consists of a long sequence of one-dimensional samples, typically sampled at 16KHz or more, which cannot be easily input into a sequence-based deep neural network model, such as an RNN or a Transformer. Traditional approaches, such as the ones used in common speech synthesis models, such as Tacotron (Wang et al, 2017)(6) and DeepVoice3 (Ping et al, 2017)(7), speech recognition models, such as AISHELL-1 (Bu et al, 2017)(8), Transducer ASR (Raju et al, 2021)(9) and many others use traditional spectrograms or FFT filter banks. In recent years, more adaptive approaches have introduced, based on foundational models pretrained in an unsupervised or semisupervised fashion on large, unlabeled datasets. Some notable unsupervised speech models include the WaveNet Autoencoder (Chorowski et al, 2019)(10), Wav2vec (Schneider et al, 2019)(11) Wav2Vec2 (Baevski et al, 2020)(12) learn a low-dimensional representation of audio in a self-supervised way by encoding raw audio into a compact latent space and then using a contrastive approach of identifying true latent representations from distractors. The subject of this study involves the use of unsupervised representations in a variety of speech processing tasks. For example, in Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition (Zhang et al, 2020)(13) wav2vec in combination with Conformer is used to achieve state-of-the-art results in transfer learning.

Prior work exists on the study of scaling of acoustic models; for instance, Scaling Laws for Acoustic Models (Droppo and Elibol, 2021)(14) studies the scaling properties of the acoustic model itself, rather than downstream tasks. The Transformer in Action study (Wang et al, 2020)(15) compares the performance of various acoustic model but does not derive a scaling law.

3 Methods

3.1 Wav2vec2 Model

The study of scaling laws on the downstream task Automated Speech Recognition is done on the model Wav2vec2 (12). It is a self-supervised model that learns representations from the raw audio provided as input. The input is first passed through a feature encoder which is made up of temporal convolutions, layer normalisation, and activation function where the raw audio is normalized to zero mean and unit variance. the encoder computes the number of time-steps for the transformer. This output is fed to a context network that gives us contextualized representations which contains relative positional embedding. The contextualized representations are then discretized via product quantization and get quantized representations. A certain part of the representation is masked and then fed to the transformer with the sim of recognizing the correct quantized latent representation that is masked, this results in contextualized representations. The working of wav2vec can be seen in Figure 1 . wav2vec follows the concept of contrastive learning, where the contextualized representation generated must match the masked quantized representation for a set of distractors K. The loss is a combination of contrastive loss and diversity loss and is described as,

$$\mathcal{L} = \mathcal{L}_m + \alpha\mathcal{L}_d$$

Where $\mathcal{L}_m$ is the contrastive loss and is represented as,

$$\mathcal{L}_m = -log\frac{\exp{(sim(c_t, q_t)/\mathcal{K})}}{\sum_{\hat{q}\sim Q_t}\exp{(sim(c_t, \hat{q})/\mathcal{K})}}$$

where, $sim(a, b) = a_T b/||a||||b||$ is the cosine similarity between the context and quantized speech representations.
Furthermore, during experimentation wav2vec2 is refereed to as wav2vec for simplicity.

3.2 Multimodal Embeddings

This experement is inspired by the work from Amazon titled Tie Your Embeddings Down (Agrawal et al, 2020) (5), which aims to create a common vector embedding across multiple modalities. This

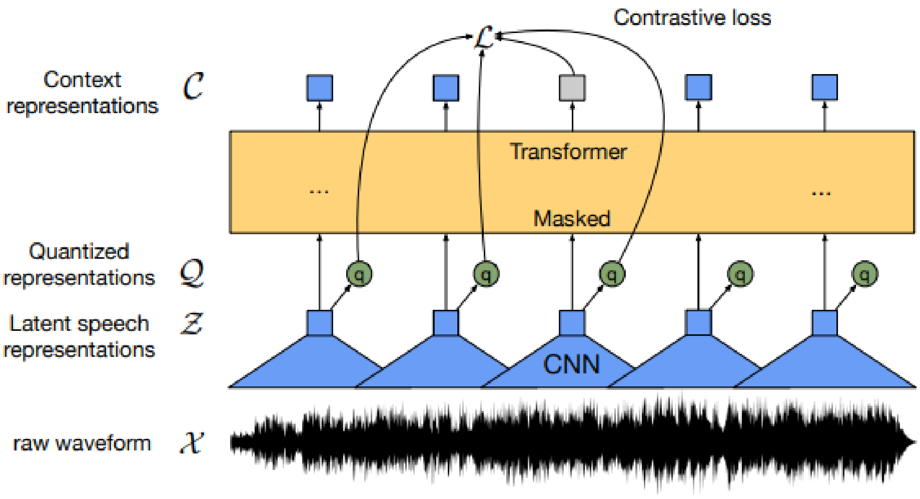
Contrastive loss
$\mathcal{L}$
Context representations $\mathcal{C}$
Transformer
...
Masked
Quantized representations $\mathcal{Q}$
q
Latent speech representations $\mathcal{Z}$
CNN
raw waveform $\mathcal{X}$

Figure 1: wav2vec Architecture

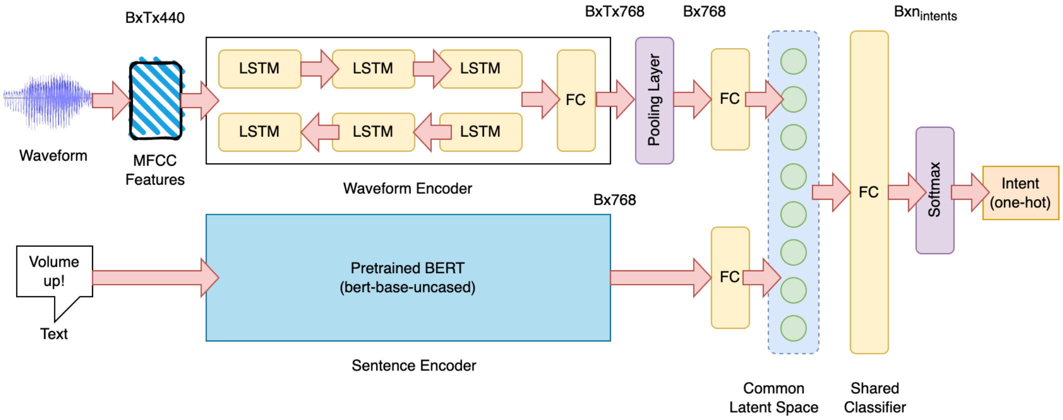

Figure 2: Tie Your Embeddings Down: Model Architecture

is achieved by combining an audio encoder with a pretrained text encoder into a single model during training with an output of the same dimension and a shared classifier and using a contrastive Triplet Loss to encourage text and audio emeeddings to be geometrically close in the latent space, as measured by the L2 distance metric, while driving both the audio and text embeddings of unrelated samples further apart. The model is summarized in Figure 2. Some possibles long-term advantage of this direction is that if one is successful at learning joint text and audio embeddings, one can then use them for zero-shot or few-shot learning - and that one obtains a representation that captures both acoustics and semantics - and relates them.

The task being considered is that of intent classification: predicting the command the user wants to execute in a personal assistant based on voice input, a rudimentary form of Natural Language Understanding (NLU). NLU has many practical applications in digital voice assistants and home automation, and production-grade NLU systems typically produce a combination of an intent and structural output with parameters to the task. For example, the intent could be a single label indicating that the user wants to make an appointment ("e.g. calendar.appointment) and structured output indicating task parameters (e.g. the date, the type of the appointment, etc), typically represented on a sequence of tokens. The focus of the present work is limited to the intent classification part.

One notable constraint of NLU training is that it requires a specially designated, manually labelled dataset annotated by humans, and the availability of such data is rather limited compared to raw,

Hyperparameter	**Value(s)**
MFCC	40 MFCCs, 80 MELs, no deltas
LSTM - hidden layer width	128, 512, 768
Transformer - Model Dimension	512
m (triplet loss margin)	64
$\lambda_{\text{audio}}/\lambda_{\text{text}}/\lambda_{\text{emb}}$	2.0 / 1.0 / 0.05
Dropout	0.2, 0.4

Table 1: Common Latent Space Model - Hyperparameters

unlabeled audio recordings. We study the effect of pretrained representations on the scaling behaviour of this model with the amount of available data.

The canonical model proposed by Agrawal et al(5) uses a simple Bi-LSTM encoder without attention with Mel-frequency cepstral coefficient features as inputs. We create several variations of the model a baseline model using the same features as Agrawal et al, and one using features from pretrained unsupervised models (RAVE(16) and wav2vec2(12). For the text embedding, we use a pretrained BERT(17) model: `bert-base-uncased` as provided by huggingface.

The desired distribution of audio and text embeddings within the common embedding space is enforced using a contrastive triplet loss.

$$\begin{aligned}\mathcal{L}_E\left(\mathbf{x}_{\text{audio}}, \mathbf{x}_{\text{text}_+}, \mathbf{x}_{\text{text}_-}\right) &= \max\left\{0, m + d(\mathbf{x}_{\text{audio}}, \mathbf{x}_{\text{text}_+}) - d(\mathbf{x}_{\text{audio}}, \mathbf{x}_{\text{text}_-})\right\} \\ \mathcal{L}_{\text{emb}} &= \text{the embedding loss} \\ \mathbf{x}_{\text{audio}} &= \text{the audio embedding} \\ \mathbf{x}_{\text{text}_+} &= \text{the positive audio example (same utterance)} \\ \mathbf{x}_{\text{text}_-} &= \text{the positive audio example (different utterance)} \\ d(\mathbf{x_1}, \mathbf{x_2}) &= \text{a distance metric between two vectors} \\ m &= \text{the margin (a hyperparameter)}\end{aligned}$$

The loss encourages the model to yield embeddings in audio samples are close to the corresponding text samples but are far from text samples with other intents in the latent space.

The following combined loss is used in training:

$$\begin{aligned}\mathcal{L}_{\text{combined}} &= \lambda_{\text{audio}}\mathcal{L}_{\text{NLL}}(\hat{\mathbf{y}}_{\text{audio}}, \mathbf{y}_{\text{audio}}) + \lambda_{\text{text}}\mathcal{L}_{\text{NLL}}(\hat{\mathbf{y}}_{\text{audio}}, \mathbf{y}_{\text{text}}) + \lambda_{\text{emb}}\mathcal{L}_{\text{emb}}\left(\mathbf{x}_{\text{audio}}, \mathbf{x}_{\text{text}_+}, \mathbf{x}_{\text{text}_-}\right) \\ \lambda_{\text{NLL}}(\hat{y}, y) &= \text{the negative likelihood loss} \\ \lambda_{\text{audio}}, \lambda_{\text{text}}, \lambda_{\text{emb}} &= \text{the weights of the audio loss, the text loss and the embedding loss, respectively}\end{aligned}$$

Table 1 shows the hyperparameters used in the experiments.

3.2.1 Global Attentional Pooling

The model proposed by Agrawal et al uses a simple Max Pooling layer over the time dimension to convert the encoder outputs to a single embedding vector. In addition to the original approach, we attempt to apply Global Attentional Pooling inspired by the attention layer used in Gated Graph Sequence Neural Networks (Li et al, 2015) (18) and its implementation in the Deep Graph Networks (DGL)(19) library as follows:

$$\begin{aligned}\mathbf{a} \in \mathbb{R}^{\text{time}} &= \text{the attention (weighting) vector} \\ \mathbf{O} \in \mathbb{R}^{\text{time}\times\text{features}} &= \text{encoder output} \\ FC(x) &= \text{a fully connected layer} \\ E_p(x) &= \text{positional embeddings(3)}\mathbf{a} \qquad\qquad = \text{softmax}(FC(\mathbf{O} + E_p(\mathbf{O}))) \\ \mathbf{o}_{\text{pool}} &= \sum_{i=1^T} a_i \mathbf{O}_{t,:}\end{aligned}$$

3.2.2 Multimodal Embeddings with Pretrained Representations

The goal of this experiment is to replace MFCC features used in the canonical model proposed by Agrawal et al with features from a pretrained model. We use features from a pre-trained RAVE(16) and Wav2Vec2(12) models. We pre-train RAVE on the English-only, multispeaker VCTK(20) dataset using its official implementation(21) using canonical hyperparameters: a latent space dimension of 128, a capacity of 64 for the encoder, 16 for the discriminator, 5 noise bands. We pretrain RAVE for 98 epochs.

For RAVE, we study the effect of allowing the fine-tuning of the acoustic model itself in the process of training the NLU model on scaling behaviour.

4 Experiments

4.1 ASR Model Scaling

The factors chosen to be scaled for the wav2vec2's performance during Automated Speech Recognition are,

- Width- Number of neurons in a model.
- Height- Number of layers in a model.
- Size of dataset.

The baseline model has width of 1024 and 2 DNN layers are used, the dataset taken is LibriSpeech English containing 100 full sentences. The scaling with respect to the width and height is by doubling or tripling these factors. The dataset is scaled by first using 100 full sentences, then using 360 full sentences and finally 460 full sentences.

4.2 NLU Model Scaling

We train the model on the Fluent Speech Commands(22), which contains 23,132 utterances by 77 different speakers for the training dataset and 3,118 and 3,793 samples for the validation and test dataset, respectively. We apply balanced sampling to ensure that every intent label has an equal probability of being selected.

We replicate the model within the SpeechBrain framework and then train each model configuration on samples of the dataset in increments of a factor of 2 (full dataset, 1/2, 1/4, 1/8, 1/16, 1/32) and take note of the best performance on a holdout set. For RAVE, we conduct the scaling experiments width widths of 128 and 512 units, with and without Global Attentional Pooling. For wav2vec, we train a model with 512 units (given it has a higher-dimensional embedding space), without Global Attentional pooling. We also attempt to replace the Bi-LSTM encoder with a 3-layer encoder-only Transformer with an encoder hidden dimension of 512.

5 Results

5.1 Scaling experiments on Downstream tasks

5.1.1 Automated Speech Recognition

The scaling experiments conducted on wav2vec2(12) were specific to a particular downstream task, Automated Speech Recognition (ASR). The aim of the task is to provide an audio input such as a statement and to output the statement in the form of text.
Different factors of the model were scaled to experiment it's effect on performance of this task. The evaluation of the model's performance during ASR can be evaluated by analysing the Word Error Rate (WER).

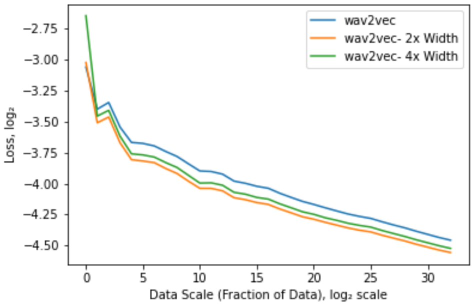

Figure 3: Width Scaling

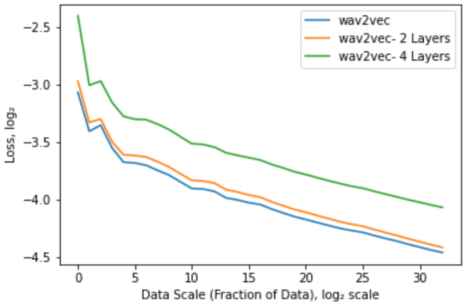

Figure 4: Layer Scaling

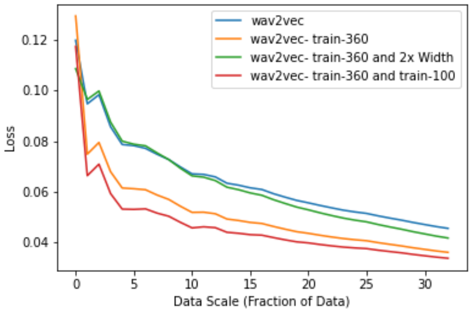

Figure 5: Scaling with respect to Dataset

5.1.2 Scaling experiments for ASR

Figure 3 represents the results of scaling wav2vec with respect to width. We observe increasing the layers, initially, performs better than the baseline model but increasing further does not improve it it's performance. It can be inferred that there is a limit to how much the width can be increased.

Figure 4 shows the results of scaling wav2vec with respect to height. The increase in height does not improve the performance of the model. It is inferred that scaling the layer of the model does not improve the model.
Figure 5 represents the results of scaling wav2vec with respect to dataset size. It can be observed that scaling data size drastically improves the performance. The model that performs the best is where 460 full sentences is used for training.

We observe the performance of these models by evaluating the metric Word Error Rate (WER). Word Error Rate is calculated by the number of errors divided by the total number of words. Table 2 displays the respective WER. IN this case, the wav2vec model trained to 460 full sentences has the lowest WER of 1.8877 indicating the best performance.

Experiment	**Word Error Rate (WER)**
wav2vec - Baseline	4.8546
wav2vec - 2 X Width	4.7516
wav2vec - 4 x Width	4.7608
wav2vec - 2 X Layer	4.8601
wav2vec - 4 X Layer	4.9979
wav2vec - train-360	2.6010
wav2vec - train-360, 2 x Width	4.852
wav2vec - train-460	1.8877

Table 2: ASR Word Error Rate

5.2 Common Latent Space Embeddings

5.2.1 Scaling Properties

Figure 6 shows the scaling behaviour of classification error with data availability for the canonical model with a common embedding space. In these figures, the data scale is represented as a fraction of the full training set being used in training.

The training set contains 23,132 speech samples. For instance, a value of 1.0 means 23,132 have been used, a value of 0.5 means 11566 samples were used, etc.

When using the Fluent Speech Commands dataset, the model shows a consistent scaling trend. The use of attentional pooling appears to have a negligible impact on the scaling behaviour, resulting in a slightly lower classification accuracy in low-data regimes compared to max-pooling without attention. The likely explanation is that global attentional pooling is an operation with learnable parameters, whereas max pooling is fixed, and the former requires data to train and is more susceptible to additional overfitting or being insufficiently trained in low-data regimes.

We empirically find that the audio classification error appears to approximately follow a classic power scaling law with respect to dataset size, of the form

$$\begin{aligned} E &= \alpha \left(\frac{D}{D_{\text{ref}}} \right)^{\beta} \\ E &= \text{the estimated scaling error} \\ \alpha, \beta &= \text{empirically derived multiplier and exponent} \\ D &= \text{the size of the dataset (in samples)} \\ D_{\text{ref}} &= \text{the size of the reference dataset} \end{aligned}$$

The following scaling laws were empirically derived:

Experiment	**Scaling Law**
Baseline - Global Attentional Pooling	$E = 0.252 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.398}$
Baseline - Max Pool	$E = 0.231 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.39}$
RAVE - Width 128 - Max Pool	$E = 0.177 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.444}$
RAVE - Width 128 - GAP	$E = 0.221 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.435}$
RAVE - Width 512 - Max Pool	$E = 0.146 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.52}$
RAVE - Width 512 - GAP	$E = 0.153 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.529}$
RAVE - Width 128 - Fine-Tuned	$E = 0.217 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.437}$
RAVE - Width 128 - Fine-Tuned On Delay	$E = 0.217 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.437}$
Transformer - Width 512	$E = 0.362 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.268}$
Transformer - Width 512, $\frac{D}{D_{\text{ref}}} \geq 0.25$	$E = 0.31 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.488}$
Wav2Vec2 - Width 512	$E = 0.063 \left(\frac{D}{D_{\text{ref}}} \right)^{-0.676}$

Table 3: Scaling Laws for NLU Models - Audio Accuracy

Figure 7 compares the empirically derived scaling trends for common latent space models using RAVE autoencoder features compared to those using canonical MFCC features. It can be seen that while the use of RAVE is beneficial in a small-data regime, it appears to have a only a modest effect on the scaling exponent. Predictably, wide (hidden layer width = 512) RAVE encoders outperform narrow ones (hidden layer width = 128), while even narrow RAVE encoders outperform wide MFCC ones,

Experiment	**Scaling Law**
Baseline	$E = 0.096 \log_2 \left(\frac{D}{D_{\text{ref}}}\right) + 0.699$
Wav2Vec2 - Width 512	$E = 0.128 \log_2 \left(\frac{D}{D_{\text{ref}}}\right) + 0.863$
RAVE - Width 512 - Max Pool	$E = 0.113 \log_2 \left(\frac{D}{D_{\text{ref}}}\right) + 0.722$

Table 4: Scaling Laws for Audio and Text Embedding Cosine Similarity - Up to D/D_{ref} = 0.5

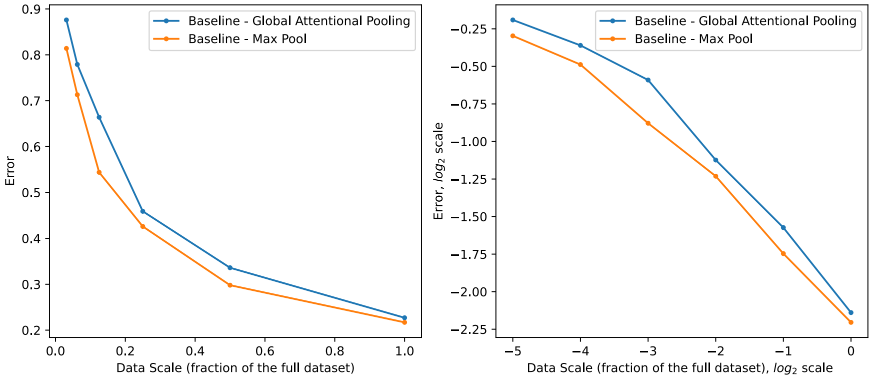

Figure 6: NLU - FSC - Common Latent Space - Audio Classification Error Scaling

which confirms the hypothesis that the self-supervised pretraining of an autoencoder is beneficial for transfer.

For the narrow RAVE model, the use of Global Attentional Pooling appears to have a slight detrimental effect on performance; however, for the wider one it appears to be partially offsetting the capacity saturation effect observed in the max pooling-based model.

Figure 8 shows a scaling comparison between a baseline LSTM-based model and a Transformer model, both trained on raw MFCC features without using a pretrained audio model.

Figure 9 shows the effect of allowing the fine-tuning of the RAVE encoder. Only a slight improvement is seen.

Figure 10 shows a comparison between a baseline model, a model with pretrained RAVE(16) representations as features and a model with pretrained wav2vec2(12) representations. Out of these models, wav2vec2(12) appears to be the most data-efficient.

5.3 NLU - Latent Space Analysis

We empirically determine that cosine similarities approximately follow a scaling law of the following form:

$$
\begin{aligned}
S &= \alpha \log_2 \left(\frac{D}{D_{\text{ref}}}\right) + \beta \\
S &= \text{the average cosine similarity between audio and text embeddings} \\
\alpha, \beta &= \text{coefficients (determined via regression)} \\
D &= \text{the size of the dataset} \\
D_{\text{ref}} &= \text{the size of the reference dataset}
\end{aligned}
$$

We derive the scaling laws shown in Table 5 for the cosine similairty between text and audio embeddings.

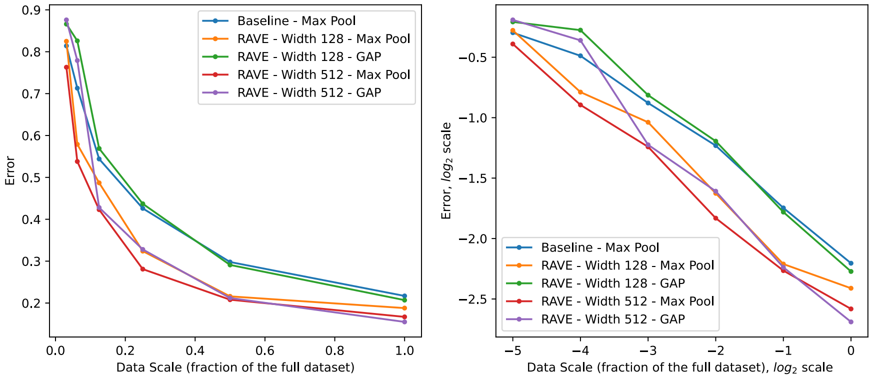

Figure 7: NLU - FSC - Common Latent Space - Audio Classification Error Scaling - RAVE

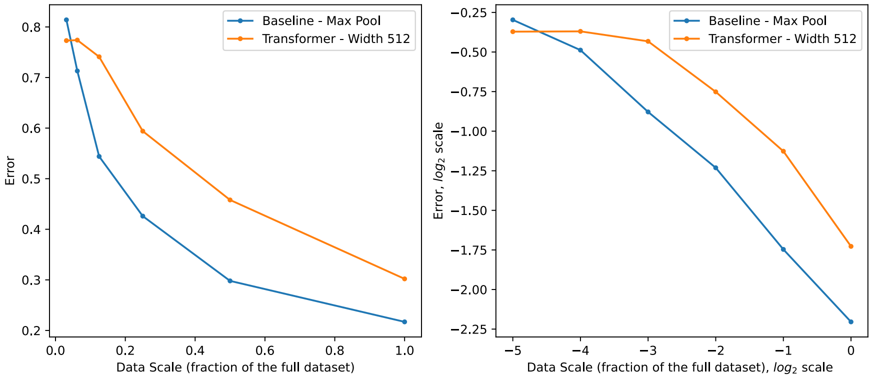

Figure 8: NLU - FSC - Common Latent Space - Scaling Properties - LSTM vs Transformer

Here we also find that wav2vec2(12) features provide the best training trajectory.

We also conduct a cursory qualitative exploration of the embedding space by performing dimensionality reduction to two dimensions using Principal Component Analysis (PCA) and visualizing the principal components. We visualize the validation latent embeddings from two pairs of labels:

- `increase.volume.none` and `activate.lights.kitchen`
- `change language.English.none` and `change language.Korean.none`

The visualizations are shown in Figure 12

The first pair consists of labels with minimal surface acoustic similarity, whereas the second pair is expected to be highly similar. We observe that predictably, in the first case, the audio embeddings occupy separate regions of the embedding space with minimal overlap, whereas the for the second pair, the audio samples of both labels are scattered in the same area.

Also, in both cases, we observe that the text embeddings of a given class are highly concentrated in a small region of the embedding space, whereas the audio samples have a much higher variance and thus are scattered around a much larger region.

We find that the embedding cosine similarity shows a positive correlation with audio accuracy ($r^2 = 0.34$) for a model trained from scratch, whereas for a wav2vec2(12)-based model, it shows virtually no correlation.

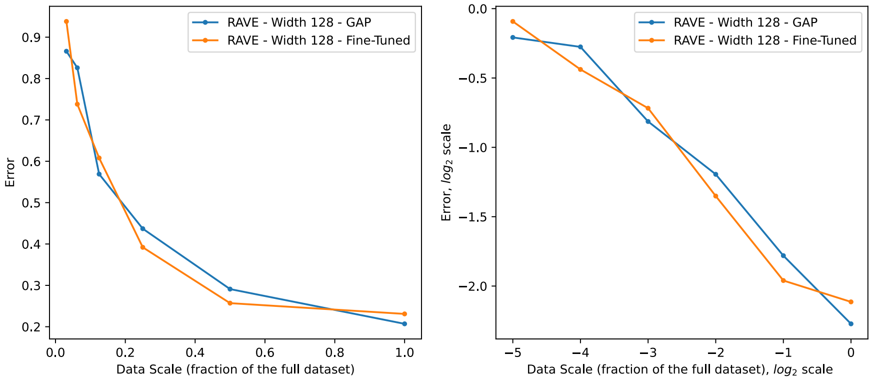

Figure 9: NLU - FSC - Common Latent Space - Effects of Fine-Tuning RAVE

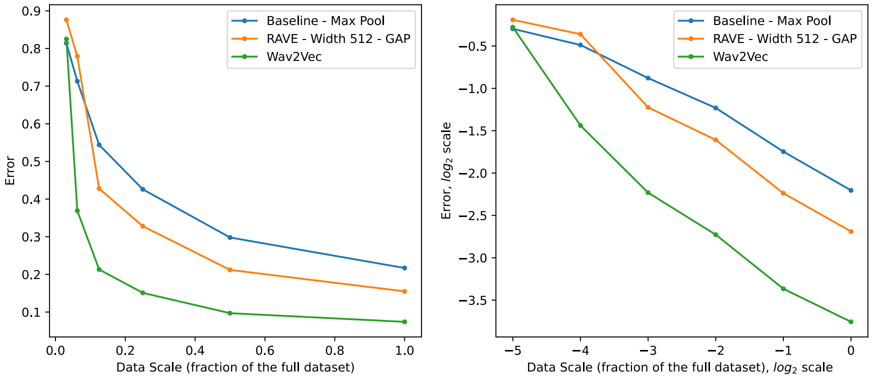

Figure 10: NLU - FSC - Common Latent Space - Scaling Properties - Baseline vs RAVE vs Wav2Vec

6 Discussion

The experiments in the present work have shown that the use of pretrained representations is beneficial in low-data regimes. We observe that in the case of RAVE, allowing the parameters of the base model to be updated improves the scaling exponent only slightly; it is not clear whether the effect is sustained.

At a similar scale, the transformer encoder appears to be less data-efficient than the canonical LSTM RNN-based encoder proposed by Agrawal, et al(5). Neither model underwent an effective hyperparameter search or a search of activations, normalization methods etc - while it might be possible to construct a Tie Your Embeddings Down model with similar performance to LSTM with an extensive hyperparameter/architecture search, this particular study did not show an immediate advantage to using a shallow, encoder-only Transformer. Also, we observe that when regression is computed over only datapoints using 25% or more of the dataset where a more stable linear trend is observed, the Transformer has a similar scaling exponent is comparable to the baseline model, which suggests that the Transformer has higher initial data requirements to achieve adequate performance but could scale reasonably well with more data.

The common latent space model appears to be achieving the desired goal of making text and audio embeddings occupy similar regions in the latent space; however, audio embeddings suffer from high variability compared to text embeddings, particularly for examples that with significant overlap in auditory features. This is likely partly due to audio features being very high-dimensional compared to their textual counterparts and containing features that are semantically irrelevant to the classification

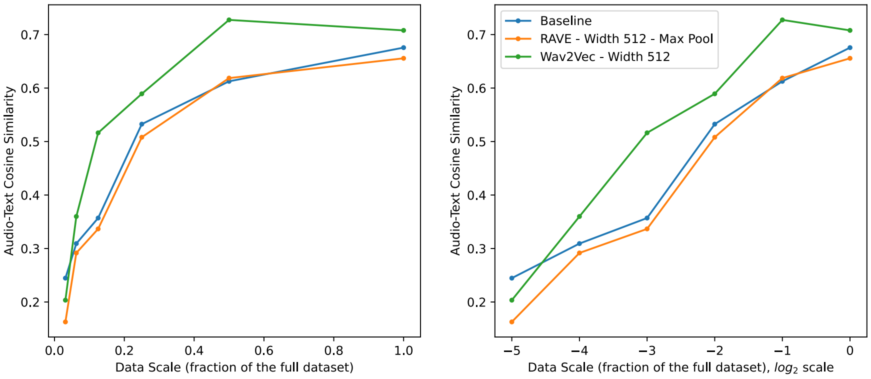

Figure 11: NLU - FSC - Common Latent Space - Audio-Text Cosine Similarity

Experiment	**Scaling Law**
Baseline	$E = 0.093 \log_2\left(\frac{D}{D_{\text{ref}}}\right) + 0.687$
Wav2Vec - Width 512	$E = 0.106 \log_2\left(\frac{D}{D_{\text{ref}}}\right) + 0.781$
RAVE - Width 512 - Max Pool	$E = 0.103 \log_2\left(\frac{D}{D_{\text{ref}}}\right) + 0.687$

Table 5: Scaling Laws for Cosine Similarity - Full Range

task, such as voice timber, accent, etc rhythm, etc. However, this also indicates that the encoding scheme could benefit from additional improvement to reduce the variance, particularly given that overfitting has been observed during training.

Future work on this subject may include studying the scaling behaviours of these models in a continual learning context with regards to speakers, languages, etc, as well as extending the NLU model to more complex scenarios representative of production deployments, as well as zero-shot and few-shot learning with previously unseen data and more complex bimodal text and speech representation learning with a time dimension.

The scaling experiments in ASR on wav2vec2 shows that data scaling works far better than scaling by width or height. In comparison to width vs height, width models scaled better while layer scaling showed to perform worse as the layers kept increasing.

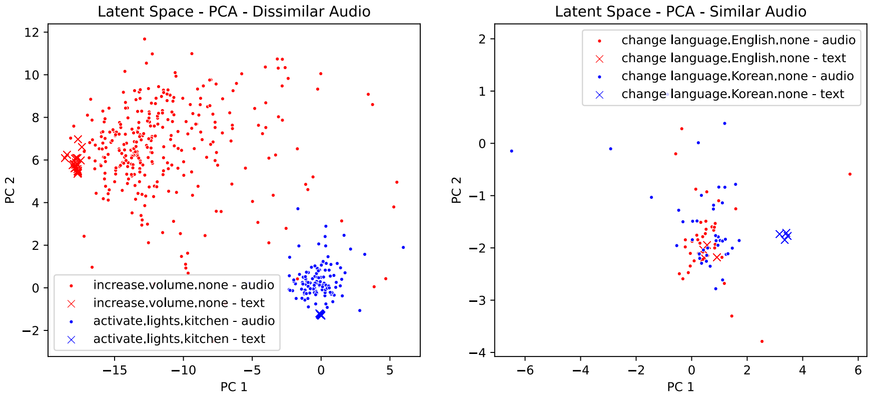

Figure 12: Latent Space Exploration - Audio and Text Embeddings

In regards to future work in ASR may include studying the behaviour zero shot learning with different datasets in English such as CommonVoice and SLURP. Also other properties of the model such as transfer learning with different languages such as Italian and French.

7 Conclusion

Both the Wav2Vec2(12)-based Automatic Speech Recognition (ASR) model and the common text and audio latent space(5) NLU model studied appear to follow predictable scaling laws with respect to data for the accuracy measurements used. In the case of wav2vec-based ASR, there appears to be an optimal model width and number of layers below or above width performance decreases. In the case of the common latent space, we find that the average cosine similarity between text and audio embeddings follows a logarithmic scaling laws with respect to data, whereas accuracy follows a classic power law. We also find that models based on pretrained acoustic models scale better than those trained from scratch with (12) offering the biggest advantage. The study showed no clear benefit to the use of Global Attentional Pooling or Transformers with NLU.

References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

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

[4] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. De Mori, and Y. Bengio, "Speechbrain: A general-purpose speech toolkit," 2021. [Online]. Available: https://arxiv.org/abs/2106.04624

[5] B. Agrawal, M. Müller, M. Radfar, S. Choudhary, A. Mouchtaris, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," 2020. [Online]. Available: https://arxiv.org/abs/2011.09044

[6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," 2017. [Online]. Available: https://arxiv.org/abs/1703.10135

[7] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," 2017. [Online]. Available: https://arxiv.org/abs/1710.07654

[8] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: an open-source mandarin speech corpus and A speech recognition baseline," *CoRR*, vol. abs/1709.05522, 2017. [Online]. Available: http://arxiv.org/abs/1709.05522

[9] A. Raju, G. Tiwari, M. Rao, P. Dheram, B. Anderson, Z. Zhang, B. Bui, and A. Rastrow, "End-to-end spoken language understanding using rnn-transducer asr," 2021. [Online]. Available: https://arxiv.org/abs/2106.15919

[10] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, dec 2019. [Online]. Available: https://doi.org/10.1109%2Ftaslp.2019.2938863

[11] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *CoRR*, vol. abs/1904.05862, 2019. [Online]. Available: http://arxiv.org/abs/1904.05862

[12] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[13] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," 2020. [Online]. Available: https://arxiv.org/abs/2010.10504

[14] J. Droppo and O. Elibol, "Scaling laws for acoustic models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09488

[15] Y. Wang, Y. Shi, F. Zhang, C. Wu, J. Chan, C.-F. Yeh, and A. Xiao, "Transformer in action: a comparative study of transformer-based acoustic models for large scale speech recognition applications," 2020. [Online]. Available: https://arxiv.org/abs/2010.14665

[16] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *CoRR*, vol. abs/2111.05011, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[18] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, cite arxiv:1511.05493Comment: Published as a conference paper in ICLR 2016. Fixed a typo. [Online]. Available: http://arxiv.org/abs/1511.05493

[19] "Dgl: Deep graph library," https://github.com/dmlc/dgl, 2022.

[20] K. M. Junichi Yamagishi, Christophe Veaux, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92), [sound], year = 2019, institution = University of Edinburgh. The Centre for Speech Technology Research (CSTR), howpublished= https://doi.org/10.7488/ds/2645."

[21] A. Caillon and P. Esling, "Rave: Realtime audio variational autoencoder," https://github.com/acids-ircam/RAVE, 2022.

[22] fluent.ai, "Fluent speech commands: A dataset for spoken language understanding research," Tech. Rep., 2020.