Effectiveness of self-supervised pre-training for speech recognition

Alexei Baevski, Michael Auli, Abdelrahman Mohamed
Facebook AI Research
{abaevski, michaelauli, abdo}@fb.com

Abstract

We present pre-training approaches for self-supervised representation learning of speech data. A BERT, masked language model, loss on discrete features is compared with an InfoNCE-based constrastive loss on continuous speech features. The pre-trained models are then fine-tuned with a Connectionist Temporal Classification (CTC) loss to predict target character sequences. To study impact of stacking multiple feature learning modules trained using different self-supervised loss functions, we test the discrete and continuous BERT pre-training approaches on spectral features and on learned acoustic representations, showing synergetic behaviour between acoustically motivated and masked language model loss functions. In low-resource conditions using only 10 hours of labeled data, we achieve Word Error Rates (WER) of 10.2% and 23.5% on the standard test “clean” and “other” benchmarks of the Librispeech dataset, which is almost on par with previously published work that uses 10 times more labeled data. Moreover, compared to previous work that uses two models in tandem (Baevski et al., 2019b), by using one model for both BERT pre-training and fine-tuning, our model provides an average relative WER reduction of 9%.1

1 Introduction

Representation learning has been an active research area for more than 30 years (Hinton et al., 1986), with the goal of learning high level representations which separates different explanatory factors of the phenomena represented by the input data (LeCun et al., 2015; Bengio et al., 2013). Disentangled representations provide models with exponentially higher ability to generalize, using little amount of labels, to new conditions by combining multiple sources of variations.

Building Automatic Speech Recognition (ASR) systems, for example, requires a large volume of training data to represent different factors contributing to the creation of speech signals, e.g. background noise, recording channel, speaker identity, accent, emotional state, topic under discussion, and the language used in communication. The practical need for building ASR systems for new conditions with limited resources spurred a lot of work focused on unsupervised speech recognition and representation learning (Park and Glass, 2008; Jansen et al.; Glass; et. al., a,f; Chung and Glass; van den Oord et al., 2018; Schneider et al.), in addition to semi- and weakly-supervised learning techniques aiming at reducing the supervised data needed in real-world scenarios (Vesely et al.; Li et al., b; Krishnan Parthasarathi and Strom; Li et al., a; Chrupała et al.; Kamper et al., 2017).

Recently impressive results have been reported for representation learning, that generalizes to different downstream tasks, through self-supervised learning for text and speech (Devlin et al., 2018; Baevski et al., 2019a; van den Oord et al., 2018; Schneider et al.; Baevski et al., 2019b). Self-supervised representation learning is done through tasks to predict masked parts of the input, reconstruct inputs through low bit-rate channels, or contrast similar data points against different ones. Different from (Baevski et al., 2019b) where the a BERT-like model is trained with the masked language model loss, frozen, and then used as a feature extractor in tandem with a final fully supervised convolutional ASR model (Collobert et al., 2016), in this work, our “Discrete BERT” approach achieves an average relative Word Error Rate (WER) reduction of 9% by pre-training and fine-tuning the same BERT model using a Connectionist Temporal Classification (Graves et al.) loss.

In addition, we present a new approach for pre-training bi-directional transformer models on
continuous speech data using the InfoNCE loss (van den Oord et al., 2018) – dubbed “continuous BERT”.

To understand the nature of their learned representations, we train models using the continuous and the discrete BERT approaches on spectral features, e.g. Mel-frequency cepstral coefficients (MFCC), as well as on pre-trained Wav2vec features (Schneider et al., 2019). These comparisons provide insights on how complementary the acoustically motivated contrastive loss function is to the other masked language model one.

The unsupervised and semi-supervised ASR approaches is in need for test suites like the unified downstream tasks available for language representation models (Devlin et al., 2018). (Irie et al., 2019; Kahn et al., 2019; Lscher et al., 2019) evaluated semi-supervised self-labeling WER performance on the standard test “clean” and test “other” benchmarks of the Librispeech dataset (Panayotov et al., 2015) when using only 100 hour subset as labeled data. (Schneider et al., 2019; Baevski et al., 2019b; van den Oord et al., 2018) use the same 960h Librispeech data as unlabeled pre-training data; however, they use Phone Error Rates (PER) on the 3h TIMIT dataset (Garofolo et al., 1993) as their performance metric. The zero-resource ASR literature (et. al., a; Chaabouni et al.) use the ABX task to evaluate the quality of learned features. To combine the best of these evaluation approaches, we pre-train our models on the unlabeled 960h Librispeech data with a close-to-zero supervised condition on both the left and right context of data. For training, it uses both a masked language model loss, by randomly removing some input words for the model to predict, and a contrastive loss to distinguish the next sentence in the document from a randomly selected one.

2.2 Wav2Vec

Wav2vec (Schneider et al., 2019) learns representations of audio data by solving a self-supervised context-prediction task with the same loss function as word2vec (Mikolov et al., 2013; van den Oord et al., 2018). The model is based on two convolutional neural networks where the encoder $f : \mathcal{X} \rightarrow \mathcal{Z}$ produces a representation $z_i$ for each time step $i$ at a rate of 100 Hz and the aggregator $g : \mathcal{Z} \rightarrow \mathcal{C}$ combines multiple encoder time steps into a new representation $c_i$ for each time step $i$. Given $c_i$, the model is trained to distinguish a sample $z_{i+k}$ that is $k$ steps in the future from distractor samples $\tilde{z}$ drawn from a distribution $p_n$, by minimizing the contrastive loss for steps $k = 1, \ldots, K$:

$$
L_k = - \sum_{i=1}^{T-k} \left( \log \sigma(z_{i+k}^\top h_k(c_i)) + \lambda \mathbb{E}_{\tilde{z} \sim p_n} [\log \sigma(-\tilde{z}^\top h_k(c_i))] \right)
$$

where $T$ is the sequence length, $\sigma(x) = 1/(1 + \exp(-x))$, and $\sigma(z_{i+k}^\top h_k(c_i))$ is the probability of $z_{i+k}$ being the true sample. A step-specific affine transformation $h_k(c_i) = W_k c_i + b_k$ is applied to $c_i$ (van den Oord et al., 2018). The loss $\mathcal{L} = \sum_{k=1}^{K} L_k$ is optimized by summing (1) over different step sizes. The learned high level features produced by the context network $c_i$ are shown to be better acoustic representations for speech recognition compared to standard spectral features.

2.3 VQ-Wav2vec

VQ-Wav2vec (Baevski et al., 2019b) learns vector quantized (VQ) representations of audio data using a future time-step prediction task. Similar to wav2vec, there is a convolutional encoder and decoder networks $f : \mathcal{X} \rightarrow \mathcal{Z}$ and $g : \mathcal{Z} \rightarrow \mathcal{C}$ for feature extraction and aggregation. However, in between them there is a quantization module $q : \mathcal{Z} \rightarrow \hat{\mathcal{Z}}$ to build discrete representations which are input to the aggregator.

First, 30ms segments of raw speech are mapped to a dense feature representation $z$ at a stride of 10ms using the encoder $f$. Next, the quantizer $q$ turns these dense representations into discrete indices which are mapped to a reconstruction $\hat{z}$ of the original representation $z$. The $\hat{z}$ is fed into the
aggregator \( g \) and the model is optimized via the same context prediction task as wav2vec (cf. §2.2). The quantization module replaces the original representation \( z \) by \( \hat{z} = e_i \) from a fixed size codebook \( e \in \mathbb{R}^{V \times d} \) which contains \( V \) representations of size \( d \).

3 Approach

3.1 Discrete BERT

Our work builds on the recently proposed work in (Baevski et al., 2019b) where audio is quantized using a contrastive loss, then features learned on top by a BERT model (Devlin et al., 2018). For the vq-wav2vec quantization, we use the gumbel-softmax vq-wav2vec model with the same setup as described in (Baevski et al., 2019b). This model quantizes the Librispeech dataset into 13.5k unique codes.

To understand the impact of acoustic representations baked into the wav2vec features, as alternatives, we explore quantizing the standard mel-frequency cepstral coefficients (MFCC) and log-mel filterbanks coefficients (FBANK), choosing a subset small enough to fit into GPU memory and running k-means with 13.5k centroids (to match the vq-wav2vec setup) to convergence. We then assign the index of the closest centroid to represent each time-step.

We train a standard BERT model (Devlin et al., 2018; Liu et al., 2019) with only the masked language modeling task on each set of inputs in the same way as described in (Baevski et al., 2019b), namely by choosing tokens for masking with probability of 0.05, expanding each chosen token to a span of 10 masked tokens (spans may overlap) and then computing a cross-entropy loss which attempts to maximize the likelihood of predicting the true token for each one that was masked (Figure 1a).

3.2 Continuous BERT

A masked language modeling task cannot be performed with continuous inputs and outputs, as there are no targets to predict in place of the masked tokens. Instead of reconstructing the input as in (van den Oord et al., 2017), we classify the masked positive example among a set of negatives. The inputs to the model are dense wav2vec features (Schneider et al., 2019), MFCC or FBANK features representing 10ms of audio data. Some of these inputs are replaced with a mask embedding and are then fed into a transformer encoder. We then compute the dot product between the outputs corresponding to each masked input, the true input that was masked, and a set of negatives sampled from other masked inputs within the same batch. The model is optimized with the InfoNCE loss (van den Oord et al., 2018) where given one positive sample \( z_i \) and \( N \) negative samples \( \tilde{z} \) we minimize:

\[
L_k = \sum_{i=1}^{T} \frac{\exp(z_i)}{\sum_{j=1}^{N} \exp(\tilde{z}_j)}
\]

where each sample \( z_i \) is computed as a dot product of the output of the model at timestep \( i \) and the true unmasked value of positive example at timestep \( i \) or a randomly sampled negative example. To stabilize training, we add the squared sum of logits produced by the dot-product to the loss, and then apply a soft clamp \( \hat{s}_i = \lambda \tanh(s_i / \lambda) \) for each logit \( s_i \) to prevent the model’s tendency to continually increase the magnitude of logits during training (Bachman et al., 2019).

3.3 Supervised fine-tuning

The pre-trained models are fine-tuned to perform the ASR task by adding a randomly initialized linear projection on top of the features computed by the transformer models into \( V \) classes representing the vocabulary of the task. The vocabulary is 29 tokens for character targets plus a word boundary token. The models are optimized by minimizing the CTC loss. Fine-tuning requires only a few epochs on a single GPU.

4 Experiments

All of our experiments are implemented by extending the fairseq (Ott et al., 2019) toolkit.

4.1 Data

All of our experiments are performed by pre-training on 960 hours of Librispeech (Panayotov et al., 2015) training set, fine-tuning on labeled 10 hours and 1 hour sets sampled equally from the two conditions of the training set, and evaluating on the standard dev and test splits.

4.2 Models

4.2.1 Quantized Inputs Training

We first train the vq-wav2vec quantization model following the gumbel-softmax recipe described in (Baevski et al., 2019b). After training this model
on 960h of Librispeech and quantizing the training dataset, we are left with 13.5k unique codewords combinations.

For quantizing MFCC and log-mel filterbanks we first compute dense features using the scripts from the Kaldi (Povey) toolkit. We then compute 13.5k K-Means centroids, to match the number of unique tokens produced by the vq-wav2vec model, using 8 32GB Volta GPUs. To fit into GPU memory, we subsample 50% of MFCC features and 25% of FBANK features from the training set before running the K-Means algorithm.

The model we use for the masked language modeling task is a standard BERT model with 12 layers, model dimension 768, inner dimension (FFN) 3072 and 12 attention heads (Devlin et al., 2018). The learning rate is warmed up over the first 10,000 updates to a peak value of $1 \times 10^{-5}$, and then linearly decayed over a total of 250k updates. We train on 128 GPUs with a batch size of 3072 tokens per GPU giving a total batch size of 393k tokens (Ott et al., 2018). Each token represents 10ms of audio data.

To mask the input sequence, we follow (Baevski et al., 2019b) and randomly sample $p = 0.05$ of all tokens to be a starting index, without replacement, and mask $M = 10$ consecutive tokens from every sampled index; spans may overlap.

4.2.2 Continuous Inputs Training

For training on dense features, we use a model similar to a standard BERT model with the same parameterization as the one used for quantized input training, but we use the wav2vec, MFCC or FBANK inputs directly. We add 128 relative positional embeddings at every multi-head attention block as formulated in (Dai et al., 2019) instead of fixed positional embeddings to ease handling longer examples. We train this model on only 8 GPUs, with a batch size of 9600 inputs per GPU resulting in a total batch size of 76,800. We find that increasing the number of GPUs (which increases the effective batch size) does not lead to better results with this particular setup.

Wav2vec features are 512-dimensional, while MFCC features have 39 dimensions and Logmel features have 80. We introduce a simple linear projection from the feature dimension to BERT dimension (768) for all models.

Similarly to the approach in 4.2.1, we choose time-steps to mask by randomly sampling, with-
out replacement, $p = 0.05$ of all time-steps to be a starting index, and mask $M = 10$ consecutive time-steps from every sampled index; spans may overlap. We sample 10 negative examples from other masked time-steps from the same example, and an additional 10 negative examples from masked time-steps occurring anywhere in the batch. We compute a dot product between the original features and the output corresponding to the same time-step after they are processed by the BERT model. We add the squared sum of logits from these computations multiplied by $\lambda = 0.04$ to the loss, and then apply a smooth clamp by recomputing each logit $\hat{s}_i = 20 \tanh(\frac{s_i}{20})$.

The learning rate is warmed up over the first 10,000 updates to a peak value of $1 \times 10^{-5}$, and then linearly decayed over a total of 250k updates.

4.3 Methodology

For quantized inputs, we compute token indices using the gumbel-softmax based vq-wav2vec model. For MFCC and FBANK features we take the index of the closest centroid (as measured by finding the minimum Euclidean distance) to each corresponding feature in the Librispeech dataset. We then train a BERT model as described in §4.2.1.

For wav2vec continuous inputs, we use features extracted by the publicly available wav2vec (Schneider et al., 2019) model which contains 6 convolutional blocks in the feature extractor and 11 convolutional blocks in the aggregator module. We use the outputs of the aggregator as features. For MFCC and FBANK, we use those features directly after applying a single linear projection to upsample them to the model dimensionality.

We fine-tune our pre-trained models on 1 or 10 hours of labelled data sampled from the Librispeech training set. We use the standard CTC loss and train for up to 20k updates. We find that the pre-trained models converge after only around 4k updates, while the models trained from scratch tend to converge much later, around 18k updates. We fine-tune all models with learning rate of 0.0001 that is linearly warmed up over the first 2k updates and then annealed following a cosine learning rate schedule over the last 18k updates. We set the dropout of the pre-trained BERT models to 0.1 and sweep on dropout of the BERT model outputs before the final projection layer over values between 0.0 and 0.4 in increments of 0.1. For each model, we choose a single best checkpoint that has the best loss on the validation set, which is a combination of dev-clean and dev-other standard Librispeech splits.

We use the publicly available wav2letter++ (Pratap et al., 2019) decoder integrated into the Fairseq framework with the official Librispeech 4-gram language model. We run a sweep on weights for language model score, word score and silence token weights for each model, where parameters are chosen randomly and evaluated on the dev-other Librispeech set. We use the weights found by these sweeps to evaluate and report results for all other splits. The sweeps are run with beam size of 250, while the final decoding is done with beam size of 1500.

The quantized BERT models have a limit of 2048 source tokens due to their use of fixed positional embeddings. During training we discard longer examples and during evaluation we discard randomly chosen tokens from each example until they are at most 2048 tokens long. We expect that increasing the size of the fixed positional embeddings, or switching to relative positional embeddings will improve performance on longer examples, but in this work we wanted to stay consistent with the setup in Baevski et al. (2019b).

The tandem model which uses the features extracted from the pre-trained BERT models is a character-based Wav2Letter setup of (Zeghidour et al., 2018) which uses seven consecutive blocks of convolutions (kernel size $5 \times 1.00 \times 10^3$ channels), followed by a PReLU nonlinearity and a dropout rate of $1 \times 10^{-1}$. The final representation is projected to a 28-dimensional probability over the vocabulary and decoded using the standard 4-gram language model following the same protocol as for the fine-tuned models.

4.4 Results

Table 1 presents WERs of different input features and pre-training methods on the standard Librispeech clean and other subsets using 10 hours and 1 hour of labeled data for fine-tuning. Compared to the two-model tandem system proposed in (Baevski et al., 2019b), which uses a the discrete BERT features to train another ASR system from scratch, our discrete BERT model provides an average of 13% and 6% of WER reduction on clean and other subsets respectively, by pre-training and fine-tuning the same BERT model on the 10h labeled set.
Table 1: WER of discrete and continuous BERT models trained on different features, and fine-tuned on 10 hours and 1 hour of labeled data.

| Model            | Input Features | dev clean | other | test clean | other |
|------------------|----------------|-----------|-------|------------|-------|
| **10 hours of labeled data** |                |           |       |            |       |
| Tandem \(^{2}\)  | Wav2vec        | 11.3      | 25.0  | 11.4       | 24.9  |
| Discrete BERT    | Wav2vec        | 9.5       | **23.4** | **10.2**   | **23.5** |
|                  | MFCC           | 16.0      | 39.3  | 16.0       | 40.7  |
|                  | FBANK          | 16.5      | 38.4  | 16.5       | 40.0  |
| Continuous BERT  | Wav2vec        | 13.3      | 31.9  | 14.0       | 35.0  |
|                  | MFCC           | 34.8      | 62.6  | 34.1       | 62.9  |
|                  | FBANK          | 26.2      | 54.2  | 27.3       | 55.1  |
| **1 hour of labeled data** |                |           |       |            |       |
| Discrete BERT    | Wav2vec        | 30.2      | 45.6  | 31.0       | 45.6  |
|                  | MFCC           | 68.5      | 79.2  | 68.8       | 80.0  |
|                  | FBANK          | 69.8      | 79.1  | 70.0       | 79.2  |
| Continuous BERT  | Wav2vec        | 25.5      | 48.0  | 26.6       | 50.7  |
|                  | MFCC           | 86.1      | 92.2  | 85.6       | 91.1  |
|                  | FBANK          | 75.9      | 88.9  | 75.7       | 88.4  |

Table 2: WER comparison with previously published (semi-)supervised baselines

| Labeled Data | LM            | test dev | other |
|--------------|---------------|----------|-------|
| Wang et al. (2019) | 960h 4-gram   | 2.6      | 5.6   |
| Kahn et al. (2019)  | 460h CnvLM | 5.93 24.07 |
| Irie et al. (2019)   | 100h None  | 12.9 35.5 |
| Panayotov et al. (2015) | 100h 4-gram | 6.59 22.52 |
| Lscher et al. (2019) | 100h 4-gram | 5.8 18.6 |
| Wav2vec +Discrete BERT (Ours) | 1h 4-gram | 31.0 45.6 |
|                     | 10h 4-gram | **10.2** 23.5 |

The wav2vec inputs represent one level of unsupervised feature discovery, which provides a better space for quantization compared to raw spectral features. The discrete BERT training augments the wav2vec features with a higher level of representation that captures the sequential structure of the full utterance through the masked language modeling loss. On the other hand, the continuous BERT training, given its contrastive InforNCE loss, can be viewed as another level of acoustic representations that captures longer range regularities. Using the MFCC and FBANK as inputs to the continuous and discrete BERT models provide insights on the synergies of different levels of acoustic and language model representations. Similar to the observations in (Mohamed et al., 2012), the FBANK features are more friendly to unsupervised local acoustic representation learning methods like continuous BERT, leading to consistent gains compared to MFCC features for both 10h and 1h sets. When using the MFCC and FBANK features for the discrete BERT training, the naive k-means clustering provides bad input acoustic centroids with nothing to benefit from the FBANK compared to the MFCC features. This shifts the entire representation learning load to the language modelling, discrete BERT component which is identical for both FBANK and MFCC, leading to almost similar performance for both input features in both the 10h and 1h fine-tuning conditions. Using the quantized wav2vec features instead provides a boost of about 40% relative improvement on average compared to the quantized FBANK features in the 10h fine-tuning case.

\(^{2}\) This is our reproduction of the tandem system in (Baevski et al., 2019b) which trains a convolutional model from scratch on features extracted of the discrete BERT model with Wav2vec input features, and evaluated on the Librispeech standard “clean” and “other” subsets.
model plays the role of a language model and input wav2vec features learns high level acoustic representations, in the very low-resource condition of 1h fine-tuning, the average relative improvement between quantized FBANK and Wav2vec inputs is larger in the “clean” subsets – 55%, which require better local acoustic representations, compared to 45% WER reduction for the noisy “other” subsets that rely more on the global language modeling capabilities.

With wav2vec features providing good acoustic representations, the discrete BERT model provides an average of about 28% relative improvement over the continuous BERT model for the 10h fine-tuning condition. We believe the reason is due to the complementary nature of the discrete BERT language modelling loss and the wav2vec acoustically motivated pre-training, as opposed to the relatively redundant acoustic pre-training losses of the continuous BERT and wav2vec. In the 1h fine-tuning case, however, better local acoustic features provide more gains in the “clean” subsets compared to the “other” ones, following the same trend of the quantized FBANK and wav2vec features under the same conditions.

Table 2 shows the competitive performance of the discrete BERT approach compared to previously published work which is fine-tuned on more than 10 times the labeled data.

4.5 Ablations

To understand the value of self-supervision in our setup, Table 3 shows WERs for both continuous and discrete input features fine-tuned from random weights, without BERT pre-training, using

| Input Features | dev clean | other clean | test clean | other |
|----------------|-----------|-------------|------------|-------|
| Discrete Input |           |             |            |       |
| Wav2vec        | 1296.1    | 1276.1      | 1344.0     | 1276.2|
| MFCC           | 93.5      | 94.5        | 93.4       | 94.2  |
| FBANK          | 100.0     | 100.0       | 100.0      | 100.0 |
| Continuous Input |         |             |            |       |
| Wav2vec        | 32.3      | 51.6        | 31.2       | 54.2  |
| MFCC           | 65.5      | 81.5        | 64.3       | 81.9  |
| FBANK          | 69.5      | 83.7        | 68.2       | 83.8  |

Table 3: WERs with no pre-training for continuous and discrete input features.

The impact of adding a second layer of acoustic representation is shown by comparing the continuous BERT model trained on top of wav2vec features versus the wav2vec model fine-tuned directly using the CTC loss – only one level of learned representations. Continuous BERT training on top of wav2vec features provides substantial gains (Table 4). Adding a second layer of representation more than halved the WER, with more gains observed in the “clean” subset as also observed in 4.4.

5 Discussion and Related Work

The success of BERT (Devlin et al., 2018) and Word2Vec (Mikolov et al., 2013) for NLP tasks motivated more research on self-supervised approaches for acoustic word embedding and unsupervised acoustic feature representation (Bengio and Heigold; Levin et al.; Chung et al., b; He et al.; Chung and Glass; Settle et al.; Schneider et al., 2019; van den Oord et al., 2018; Chung et al., a; Baevski et al., 2019b), either by predicting masked discrete or continuous input, or by contrastive prediction of neighboring or similarly sounding segments using distant supervision or proximity in the audio signal as an indication of similarity. In (Kamper et al.) a dynamic time warping alignment is used to discover similar segment pairs. Our work is inspired by the research efforts in reducing the dependence on labeled data for building ASR systems through unsupervised unit discovery and acoustic representation leaning (Park and Glass, 2008; Jansen et al.; Glass; et. al., a,f), and through multi- and cross-lingual transfer learning in low-resource

| Pre-training | dev clean | other clean | test clean | other |
|--------------|-----------|-------------|------------|-------|
| Wav2vec      | 51.1      | 71.4        | 52.6       | 71.9  |
| + Continuous BERT | 13.3  | 31.9        | 14.0       | 35.0  |

Table 4: WER the continuous BERT model and the fine-tuned wav2vec feature extraction module without any BERT variant for higher level representation learning.
conditions (et. al., c,d,b; Ghoshal et al.; Huang et al.; et. al., e), and semi-supervised learning (Vesely et al.; Li et al., b; Krishnan Parthasarathi and Strom; Li et al., a).

6 Conclusion and Future work

We presented two variations, continuous and discrete, of BERT models that are pre-trained on the librispeech 960h data and fine-tuned for speech recognition rather than used as feature extractor in tandem with another ASR system. Along with the discrete-input BERT model, we used a contrastive loss for training a continuous variant of BERT. The acoustic and language modeling roles in the system are played by the vq-wav2vec and the BERT components respectively. Our ablation experiments showed the contribution and importance of each component for final ASR performance. Our system is able to reach final WER of 10.2% and 23.5% on the standard Librispeech test clean and other sets, respectively, using only 10h of labeled data, almost matching the 100h supervised baselines. Our future directions include testing our model on 1000x larger volume of unlabeled data that is more acoustically challenging, along with multi- and cross-lingual transfer learning extensions.

References

Aren Jansen et. al. a. A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition. In ICASSP 2013.

Georg Heigold et. al. b. Multilingual acoustic models using distributed deep neural networks. In ICASSP 2013.

Haihua Xu et. al. c. Semi-supervised and cross-lingual knowledge transfer learnings for dnn hybrid acoustic models under low-resource conditions. In Interspeech 2016.

Jia Cui et. al. d. Multilingual representations for low resource speech recognition and keyword search. In ASRU 2015.

Ngoc Thang Vu et. al. e. Multilingual deep neural network based acoustic modeling for rapid language adaptation. In ICASSP 2014.

Odette Scharenborg et. al. f. Linguistic unit discovery from multi-modal inputs in unwritten languages: Summary of the "speaking rosetta" JSALT 2017 workshop. In ICASSP 2018.

Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019a. Cloze-driven pretraining of self-attention networks. arXiv, abs/1903.07785.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019b. vq-wav2vec: Self-supervised learning of discrete speech representations. arXiv preprint arXiv:1910.05453.

Samy Bengio and Georg Heigold. Word embeddings for speech recognition. In INTERSPEECH 2014.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell.

Rahma Chaabouni, Ewan Dunbar, Neil Zeghidour, and Emmanuel Dupoux. Learning weakly supervised multimodal phoneme embeddings. In Interspeech 2017.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. In ACL 2017.

Yu-An Chung and James Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In INTERSPEECH 2018.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James R. Glass. a. An unsupervised autoregressive model for speech representation learning. Interpeech 2019.

Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung yi Lee, and L. Lee. b. Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. Interspeech 2016.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. arXiv, abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. 1993. Darpa timit acoustic phonetic continuous speech corpus cdrom.

A. Ghoshal, P. Swietojanski, and S. Renals. Multilingual training of deep neural networks. In ICASSP 2013.
J. Glass. Towards unsupervised speech processing. In ISSPA 2012.

Alex Graves, Santiago Fernandez, and Faustino Gomez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In ICML 2006.

Wanjia He, Weiran Wang, and Karen Livescu. Multi-view recurrent neural acoustic word embeddings. ICLR 2016.

G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. 1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Distributed Representations.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. ICASSP 2013.

Kazuki Irie, Rohit Prabhavalkar, Anjuli Kannan, Antoine Bruguier, David Rybach, and Patrick Nguyen. 2019. On the choice of modeling unit for sequence-to-sequence speech recognition. Interspeech 2019.

Aren Jansen, Kenneth Church, and Hynek Hermansky. Towards spoken term discovery at scale with zero resources.

Jacob Kahn, Ann Lee, and Awni Hannun. 2019. Self-training for end-to-end speech recognition.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. Unsupervised neural network based feature extraction using weak top-down constraints. ICASSP 2015.

Herman Kamper, Shane Settle, Gregory Shakhnarovich, and Karen Livescu. 2017. Visually grounded learning of keyword prediction from untranscribed speech. In Interspeech.

S. Krishnan Parthasarathi and N. Strom. Lessons from building acoustic models with a million hours of speech. In ICASSP 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Christoph Lascher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schelter, and Hermann Ney. 2019. Wath asr systems for librispeech: Hybrid vs attention. Interspeech 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Proc. of NIPS.

A. Mohamed, G. Hinton, and G. Penn. 2012. Understanding how deep belief networks perform acoustic modelling. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. ArXiv.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. volume abs/1711.00937.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proc. of NAACL System Demonstrations.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In Proc. of WMT.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. ICASSP 2015.

A. Park and J. Glass. 2008. Unsupervised pattern discovery in speech. IEEE TASLP, 16(1):186–197.

Daniel et. al. Povey. The kaldi speech recognition toolkit. In ASRU 2011.

V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. 2019. Wav2letter++: A fast open-source speech recognition system. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In Interspeech 2019.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. CoRR, abs/1904.05862.
Shane Settle, Kartik Audhkhasi, Karen Livescu, and Michael Picheny. Acoustically grounded word embeddings for improved acoustics-to-word speech recognition. *ICASSP 2019.*

K. Vesely, M. Hannemann, and L. Burget. Semi-supervised training of deep neural networks. In *ASRU 2013.*

Yongqiang Wang, Abdelrahman Mohamed, Duc Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, Christian Fuegen, Geoffrey Zweig, and Michael L. Seltzer. 2019. Transformer-based acoustic modeling for hybrid speech recognition.

Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schaiz, Gabriel Synnaeve, and Emmanuel Dupoux. 2018. Learning filterbanks from raw speech for phone recognition. In *Proc. of (ICASSP).*