IMPROVED LANGUAGE IDENTIFICATION THROUGH CROSS-LINGUAL SELF-SUPERVISED LEARNING

Andros Tjandra, Diptanu Gon Choudhury, Frank Zhang, Kritika Singh, Alexis Conneau, Alexei Baevski, Assaf Sela, Yatharth Saraf, Michael Auli*

Meta AI, USA

{androstj,diptanu}@fb.com

ABSTRACT

Language identification greatly impacts the success of downstream tasks such as automatic speech recognition. Recently, self-supervised speech representations learned by wav2vec 2.0 have been shown to be very effective for a range of speech tasks. We extend previous self-supervised work on language identification by experimenting with pre-trained models which were learned on real-world unconstrained speech in multiple languages and not just on English. We show that models pre-trained on many languages perform better and enable language identification systems that require very little labeled data to perform well. Results on a 26 languages setup show that with only 10 minutes of labeled data per language, a cross-lingually pre-trained model can achieve over 89.2% accuracy.

***Index Terms*—** Language identification, self-supervised learning, pre-training, multilingual, wav2vec

1. INTRODUCTION

Automatic speech recognition (ASR) has seen large improvements through better modeling [1, 2, 3] and the use of unlabeled data [4, 5, 6, 7, 8]. Despite a sizeable body of work on multilingual speech recognition [9, 10, 11, 12, 13, 14, 15, 16, 17], the vast majority of systems are trained for a single language. However, in many real-world settings, we wish to transcribe speech data in different languages and it is crucial to route utterances to the system trained for the language at hand. Language identification (LID) is typically used to identify the language of an utterance and the accuracy of this component is crucial to prevent poor ASR performance.

Language identification has been tackled with conventional methods [18] as well as with modern neural networks [19]. Most of these approaches are trained purely with labeled data, however, unlabeled data is typically much easier to collect. Self-supervised learning leverages unlabeled data to obtain good data representations that can then be fine-tuned for a particular downstream task [20, 7, 21, 8].

Prior work on LID has explored the use of a wav2vec 2.0 model pre-trained only on English data [22]. In this paper, we extend this work by considering cross-lingually trained self-supervised models [23]. In particular, we pre-train models with a large amount of unlabeled data from many different languages and then fine-tune them with as little as 10 minutes of labeled data per language for LID to enable systems for low-resource languages. The audio data used here is sampled from public social media videos, which presents unique challenges such as a variety of speaker styles and the quality of the recordings. Moreover, our approach does not use any auxiliary features as are commonly used to improve performance. We also investigate different pooling strategies to aggregate the pre-trained context-representations for LID classification task. We modified wav2vec 2.0 to use log-mel spectrogram features as input, similar to [24]. Our experiments show strong performance using as little as ten minutes of labeled data per language compared to models trained on much larger amounts of labeled data. Furthermore, multilingual pre-trained models achieve better performance than models pre-trained only with a single language.

*Currently at Google AI, work done while at Facebook AI.

2. LOG-MEL WAV2VEC ARCHITECTURE

In this section, we describe our modifications to the original wav2vec 2.0 model architecture and the cross-lingual training strategy. A wav2vec model [8, 21] consists of multiple convolution and Transformer [25] layers and operates on top of the raw waveform.

Here, we use log-mel spectrogram [24] as the input features instead of raw waveform to improve the memory and computational efficiency. We define our input feature as $X \in \mathbb{R}^{S \times F} = [x_1, .., x_S]$ where S is the number of frames in an utterance and F is the input dimension for each frame (e.g., $F = 80$ for 80-dimensional log-mel spectrogram). From here, the input features are fed into a feature encoder which is followed by a context encoder. We present our modified wav2vec architecture in Figure 1(a).

2.1. Feature encoder

A feature encoder $f(\cdot) : X \rightarrow Z$ takes input X and outputs latent speech representations $Z = [z_1, ..., z_T]$ where $T = S/R$. Compared to original wav2vec 2.0, we replace the seven convolutional layers in the encoder with a time-stacking layer and a linear layer. Due to the quadratic cost $\mathcal{O}(n^2)$ of time and memory of Transformer layers, reducing the input length greatly improves the training and inference efficiency. The time-stacking layer is defined as a function $ts(\cdot) : \mathbb{R}^{S \times F} \rightarrow \mathbb{R}^{T \times FR}$, which concatenates R consecutive frames into a single frame and reduces sequence length by a factor of $1/R$.

2.2. Context encoder

A context encoder $g(\cdot) : Z \rightarrow C$ takes the input Z produced by the feature encoder block and outputs context representations C. The context encoder consists of a linear layer, a layer normalization [26], a convolution layer to encode relative position information [8], multiple Transformers layers [25] and another linear layer.

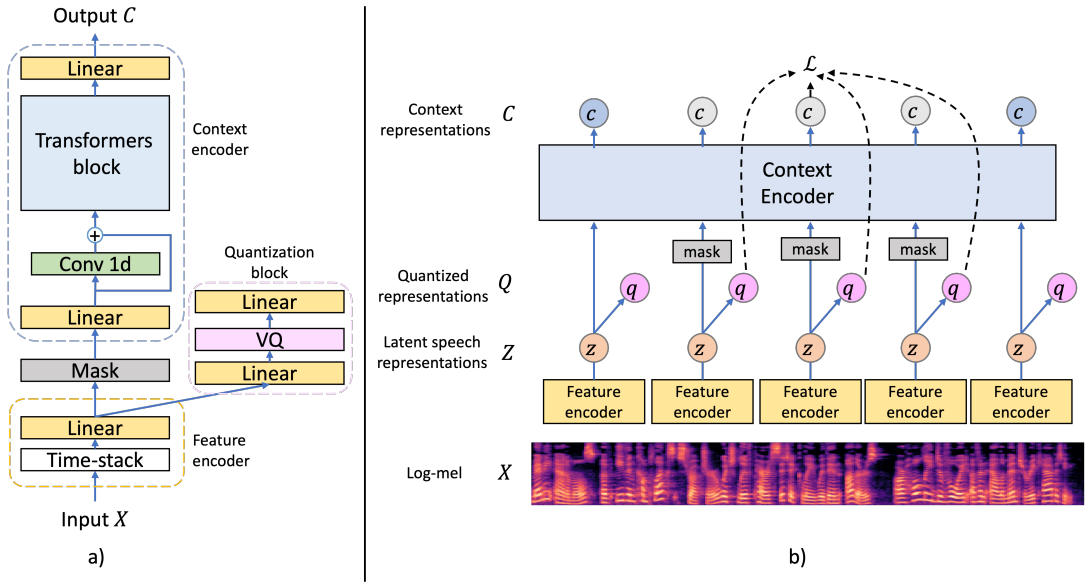

Fig. 1. a) Log-mel Wav2Vec architecture. b) An illustration of how Wav2Vec generates context representation and solves the contrastive task.

2.3. Quantization block

A quantization block $h(\cdot) : Z \rightarrow Q$ takes the output Z of the feature encoder layer and produces a quantized representation Q. A linear layer is added on top to the input Z. A product quantization [27, 28] with G groups and V codebook entries $e \in \mathbb{R}^{V \times d/G}$ was applied on the projected input. The result of each group is concatenated and another linear layer is applied to generate the quantized representation Q. A Gumbel softmax function [29] enables discretely choosing an index based on the maximum value to be fully differentiable.

3. CROSS-LINGUAL PRE-TRAINING

For multilingual pre-training, we collected large quantities of unlabeled speech from various languages and combined them into a single multilingual dataset to train cross-lingual speech representations (XLSR; [23]). The audio data used here is sampled from public social media videos, involving unconstrained, natural speech that is filled with background music and noise, with various speaking styles, disfluencies, accents, and un-cued speaker and language switching. This presents an interesting and challenging application of self-supervised learning that directly complements other recent work on self-supervised learning which focused on datasets based on audio-books, which is clean and focused on a single domain [21, 8, 24] with a few exceptions [30]. Compared to [23], we do not upsample or downsample certain languages during training because we assume no access to the language ID in the unlabeled video dataset.

Figure 1(b) illustrates how each block interacts with the other to solve the contrastive task. The model needs to find which sample is a true quantized latent q_t from a set of $K+1$ candidates $\tilde{q} \in Q$. The false quantized latent $\tilde{q} \setminus q_t$ are uniformly sampled from any masked time-step. We define this contrastive loss as:

$$\mathcal{L}_m = -\log \frac{\exp(sim(c_t, q_t))}{\sum_{\tilde{q} \sim Q} \exp(sim(c_t, \tilde{q}))} \tag{1}$$

where $sim(c_t, q_t)$ is cosine-similarity between context representations and quantized latent speech representations.

We add a diversity loss to encourage the equal usage of V codebook entries on each G codebook (Sec.2.3). The diversity loss is designed to maximize the entropy of the averaged softmax probability over the codebook entries for each codebook group $\bar{p}_g$. We define the loss as:

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -\mathbb{H}(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v} \tag{2}$$

$$\text{where} \quad \bar{p}_{g,v} = \sum_{b=1}^{B} \sum_{t=1}^{T} \frac{p_{b,t,g,v}}{BT} \tag{3}$$

and $\bar{p}_{b,t,g,v}$ is the softmax probability without gumbel noise on group g, codebook entry v, sample b, and time-step t. Our final loss is defined as $\mathcal{L} = \mathcal{L}_m + \lambda \mathcal{L}_d$ where λ is a hyperparameter to control the diversity loss.

4. LID FINETUNING

We illustrate our LID classifier architecture in Figure 2. After we pre-training a log-mel wav2Vec model, we take the latest checkpoint and use it to initialize the bottom part of an LID classifier. The context representations $C = [c_1, ..., c_T] \in \mathbb{R}^{T \times D}$ are summarized by adding a pooling function $pool(\cdot) : \mathbb{R}^{T \times D} \rightarrow \mathbb{R}^{D}$. We explore several pooling operations such as:

1. Mean pooling: $o = \sum_{t=1}^{T} c_t / T$.
2. Standard deviation pooling: $o = \sqrt{\sum_{t=1}^{T} (c_t - \mu)^2 / T}$
3. Self-attention pooling [31]:

$$a = \texttt{softmax}(w_2 \texttt{GELU}(W_1 c^{\intercal})) \in \mathbb{R}^{T} \tag{4}$$

$$o = \sum_{t=0}^{T} a_t c_t \in \mathbb{R}^{D} \tag{5}$$

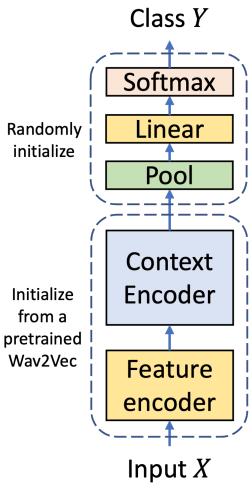

Fig. 2. A Wav2Vec encoder with pooling and softmax projection layer for utterance-level language id (LID) classification.

where $W_1 \in \mathbb{R}^{U \times D}$, $w_2 \in \mathbb{R}^U$ and o is a weighted sum from C based on attention vector a.

4. CLS-token pooling [32]: Adding a special `[CLS]` token in the beginning of sequence and set $o = c_1$.

After the pooling layer, we append a randomly initialized linear layer with L output dimensions on top of a pooled representation o, and normalize it via a softmax function. We minimize the cross-entropy loss $\mathcal{L} = -\sum_{l=1}^{L} y_l log(p_l)$ where p_l is the predicted probability of speech utterance X belonging to language l and $y_l = 1$ if l is the target class, otherwise $y_l = 0$.

5. EXPERIMENTAL SETUP

5.1. Dataset

We conducted the experiments on our in-house datasets. The training data consists of de-identified public videos with no personally identifiable information (PII), from where only the audio part is used. The dataset contains a fair amount of accented speech utterances and a good amount of local dialects for languages that are spoken across the globe. To pre-train an XLSR, we gathered up to 6.3 million hours of unlabeled audio. To pre-train a wav2vec-En, we required a large dataset with only English utterances, however, since we do not have language labels for the unlabeled video dataset, we created a 'pseudo-en' dataset by using an in-house LID model to predict the class for each unlabeled audio and pick each utterance where the predicted class is 'en'. Note, this model mostly serves as a comparison to cross-lingual pre-training with XLSR.

For the input features, we extract 80-dimensions log-mel spectrogram with 25 milliseconds window size and 10 milliseconds step size. We normalize the value for each feature dimension by subtracting it with the mean and divide by the standard deviation. The mean and standard deviation are calculated from a random subset of the pre-training dataset.

5.2. Pre-training setup

Here, we describe each module configuration inside our pre-trained wav2vec model. Inside a feature encoder, we have:

- Time-stride layer: reduce input sequence length by $R = 4$ times (80 input dimension, 320 output dimension).
- Linear layer: 320 input dimension, 512 output dimension.

Inside a context encoder, we have:

- Linear layer: 512 input dimension, 1024 output dimension.
- 1D Convolution layer: 1024 input dimension, 1024 output dimension, kernel size 48, filter groups 16.
- Transformers: 24 layers, 1024 input dimension, 16 attention head, 4096 feedforward dimension, GELU activation function, pre-layer norm [33].
- Linear layer: 1024 input dimension, 768 output dimension.

To generate quantized target Q for contrastive task, we feed latent speech representation Z into a quantization block with:

- Linear layer: 512 input dimension, 768 output dimensions.
- Gumbel VQ: 320 codebooks, 2 groups.
- Linear layer: 768 input and output dimension.

For masking over the latent speech representation Z, we sample $p = 0.065$ as the starting indices and we mask the next $M = 5$ frames. Overall, this model has 300 million parameters.

We pre-trained two models with the same architecture but different inputs:

1. wav2vec 2.0 En is trained on English only.
2. XLSR is trained with all unlabeled audios.

We set the diversity loss hyperparameter $\lambda = 0.1$ for all experiments. All models are trained using the Adam optimizer [34] with learning rate $lr = 1e-3$ for wav2vec 2.0 En and $lr = 5e-3$ for XLSR up to 300000 updates. We also add weight decay $1e-2$ with ℓ^2 weight penalty. We anneal the learning rate by using a linear decay learning schedule, with warm-up step up to 32000 updates and linearly decay to 0 after that. We crop the input sequence length up to 2000 frames (equals to 20 seconds).

5.3. Fine-tuning setup

In the finetuning step, we randomly crop the audio into a 6 seconds chunk and extract 80-dimensions log mel spectrogram. On top of a wav2vec encoder, we added a pooling layer and a linear layer with L output dimension. We prepare the finetuning dataset with 26 languages: English (en), Spanish (es), Arabic (ar), Indonesian (id), Vietnamese (vi), Portuguese (pt), Thai (th), Hindi (hi), Italian (it), French (fr), Turkish (tr), Tagalog (tl), Urdu (ur), German (de), Chinese (zh), Malayalam (ml), Bengali (bn), Russian (ru), Burmese (my), Malay (ms), Tamil (ta), Marathi (mr), Kannada (kn), Sinhalese (si), Japanese (ja), Dutch (nl). We prepare different amounts of supervised data per language: 10 minutes, 1 hour, 10 hours, 100 hours. This demonstrates the possibility of training LID on low resource scenario and improving LID result on high resource scenario on top of a self-supervised pre-trained model. All finetuning models are trained with Adam optimizer with learning rate $lr = 1e-4$, tri-stage learning rate schedule (10% warm-up step, 40% stay-step and 50% decay step), weight decay $1e-2$ with ℓ^2 weight penalty. We also have several finetuning scenarios such as from scratch (without any pre-trained stage), from a monolingual wav2vec 2.0 En checkpoint, and an XLSR checkpoint. All models have the same architecture and number of parameters.

During the finetuning training stage, we randomly sample 6 seconds audio chunks for each utterance. During the evaluation stage, we run the LID classifier on 6 seconds window and 3 seconds step size, average the language probability across multiple predictions and pick the highest probability as the prediction.

Lbl. / lang	Pre-training	Test Accuracy (%)			
		0-6s	6-18s	18-∞s	Overall
10 min	None	7.1	9.5	10.6	9.6
	w2v2 En	71.3	73.1	76.1	74.2
	XLSR	85.4	88.8	90.8	**89.2**
1 hour	None	20.2	25.2	29.5	26.5
	w2v2 En	79.3	85.9	89.3	86.5
	XLSR	87.2	92.5	94.8	**92.8**
10 hours	None	48.3	61.9	71.8	64.5
	w2v2 En	86.8	93.3	95.6	93.4
	XLSR	88.2	94.3	96.1	**94.2**
100 hours	None	72.2	84.9	90.7	86.7
	w2v2 En	89.5	95.7	97.3	95.5
	XLSR	90.3	95.9	97.2	**95.7**

Table 1. LID test accuracy for 26 languages setup using different amounts of labeled training data per language (10 minutes - 100 h). We compare three scenarios: training an LID model without pre-training (None), fine-tuning a monolingually pre-trained wav2vec 2.0 model (w2v2 En), and fine-tuning a cross-lingually pre-trained model (XLSR).

6. RESULTS

6.1. Language identification for 26 languages

The ability to train LID models with very little labeled data is important in order to be able to extend speech technology to the thousands of languages and dialects spoken around the world. The test accuracy for 26 languages experiments are calculated from a test set that contains a total of 3700 hours from 26 languages. The accuracy are shown in Table 1. We calculated the accuracy for short utterances (shorter than 6 seconds), medium utterances (6-18 seconds), long utterances (longer than 18 seconds), and overall accuracy. Based on our experiment, we show that training from scratch (None) performs particularly poorly with just 10 minutes of labeled data. On the other hand, XLSR achieves over 89.2% accuracy. Pre-training on more languages performs better with little labeled data and in the high-resource labeled data regime, the labeled data provides sufficient learning signal. There is a similar trend of training from scratch (no pre-training) improving with more labeled data but even with large amounts of labeled data, there is still a sizeable gap to pre-trained models.

6.2. Ablations

Next, we explore the effect of using representations from different Transformer blocks of the pre-trained models. For this section, we focus our experiment on 26 language setup using 10 minutes, 1 and 10 hours of labeled data per language for fine-tuning. Figure 3 shows that the middle and upper parts of the pre-trained model (from 8th to 24th layer) perform significantly better than the lower part (from 2nd to 6th layer). The result suggests that we could prune up to $2/3$ of the context encoder blocks, which reduces the time and memory usage during fine-tuning and inference while maintaining good accuracy. Additionally, by keeping only eight Transformer blocks, we reduce the number of parameters from 300 million down to 100 million.

So far we used mean pooling to aggregate the output of the pre-trained models for a given speech utterance into a single vector. Table 2 compares this strategy to max pooling, concatenated mean+max+min pooling, concatenated mean+max pooling, concatenated mean+std (statistical pooling [35]), first-step class tokens (`[CLS]`) [32], and self-attention pooling [31]. The result suggests that mean+max pooling works very well compared to the alternatives and provides a simple way to aggregate the context information.

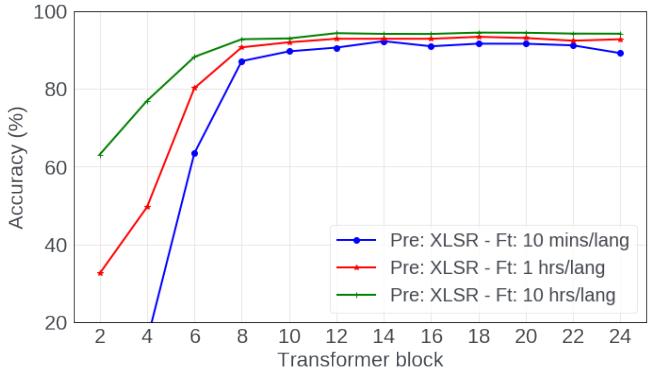

Fig. 3. LID overall accuracy when using the output of different Transformer blocks from a pre-trained (pre) XLSR model as input to the LID classifier. Accuracy is in terms of the 26 language setup and using 10 minutes, 1 and 10 hours of labeled data per language for fine-tuning (ft).

Aggregation strategy	Accuracy (%)			
	0-6s	6-18s	18-∞s	Overall
Max	86.6	92.7	94.8	92.8
Mean+Max+Min	88.1	92.9	94.7	93.0
Mean+Max	88.5	93.1	94.8	93.2
Mean+Std	84.2	90.9	93.4	91.1
`[CLS]` Token	85.4	91.4	93.9	91.7
Self Attention	87.0	92.0	94.1	92.2

Table 2. LID accuracy for different strategies to aggregate the context representations of an XLSR model on the 26 language setup using 1 hour of labeled data per language for fine-tuning.

7. CONCLUSION

In this paper, we demonstrated the benefit of using self-supervised pre-trained representations learned on unlabeled speech data to improve language identification. We showed that pre-training is more effective than training LID models solely from labeled data and cross-lingual representations are particularly effective for low-resource setups, where little labeled data is available. This is important to enabling speech technology for many more languages spoken around the world. Using only 10 minutes of labeled data per language, a cross-lingually pre-trained LID can achieve an accuracy of over 89.2% on a 26 language setup. Additionaly, we also observe the benefits of cross-lingual pre-training on LID with higher amount of labeled data. We find that we can prune up to two-thirds of the pre-trained model while achieving the same accuracy. For future work, we may explore how to make these models more efficient for inference since pre-trained models are still very large.

8. REFERENCES

[1] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. of ICASSP*, 2018.

[2] Anmol Gulati, James Qin, et al., "Conformer: Convolution-augmented transformer for speech recognition," *Proc. of Interspeech*, 2020.

[3] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, and et al., "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. of ICASSP*, 2020.

[4] Gabriel Synnaeve, Qiantong Xu, et al., "End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures," *Proc. of ICML workshop on Self-supervision in Audio and Speech (SAS)*, 2020.

[5] Qiantong Xu, Tatiana Likhomanenko, et al., "Iterative pseudo-labeling for speech recognition," in *Proc. of Interspeech*, 2020.

[6] Daniel S. Park, Yu Zhang, et al., "Improved noisy student training for automatic speech recognition," in *Proc. of Interspeech*, 2020.

[7] Yu-An Chung and James Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," in *Proc. of Interspeech*, 2018.

[8] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, vol. 33, 2020.

[9] Lukáš Burget, Petr Schwarz, et al., "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *Proc. of ICASSP*, 2010.

[10] Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee, "A study on multilingual acoustic modeling for large vocabulary asr," in *Proc. of ICASSP*, 2009.

[11] Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. of ICASSP*, 2013.

[12] Hervé Bourlard, John Dines, et al., "Current trends in multilingual speech processing," *Sadhana*, 2011.

[13] Jaejin Cho, Murali Karthick Baskar, et al., "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *Proc. of IEEE SLT*, 2018.

[14] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. of ICASSP*, 2018.

[15] Anjuli Kannan, Arindrima Datta, et al., "Large-scale multilingual speech recognition with a streaming end-to-end model," in *INTERSPEECH*, 2019.

[16] Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *Proc. of ICASSP*, 2019.

[17] Vineel Pratap, Anuroop Sriram, et al., "Massively multilingual asr: 50 languages, 1 model, 1 billion parameters," *arXiv*, vol. abs/2007.03001, 2020.

[18] Marc A Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on speech and audio processing*, 1996.

[19] Ignacio Lopez-Moreno, Javier Gonzalez-Dominguez, et al., "Automatic language identification using deep neural networks," in *Proc. of ICASSP*, 2014.

[20] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," in *Proc. of NIPS*, 2018.

[21] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Proc. Interspeech 2019*, pp. 3465–3469, 2019.

[22] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[23] 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.

[24] Yu Zhang, James Qin, et al., "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[25] Ashish Vaswani, Noam Shazeer, et al., "Attention is all you need," in *Neural Information Processing Systems 2017*, 2017, pp. 5998–6008.

[26] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *ArXiv e-prints*, pp. arXiv–1607, 2016.

[27] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *IEEE TPAMI*, vol. 33, no. 1, pp. 117–128, 2010.

[28] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR*, 2019.

[29] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*. 2017, OpenReview.net.

[30] Wei-Ning Hsu, Anuroop Sriram, et al., "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," *arXiv*, 2021.

[31] Zhouhan Lin, Minwei Feng, et al., "A structured self-attentive sentence embedding," in *ICLR*. 2017, OpenReview.net.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL 2019*, June 2019, pp. 4171–4186.

[33] Ruibin Xiong and Yunchang et al Yang, "On layer normalization in the transformer architecture," in *ICML*. PMLR, 2020, pp. 10524–10533.

[34] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] David Snyder, Garcia-Romero, et al., "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017, pp. 999–1003.