Self-Supervised Representations Improve End-to-End Speech Translation

Anne Wu, Changhan Wang, Juan Pino, Jiatao Gu

Facebook AI, USA

{annewu, changhan, juancarabina, jgu}@fb.com

Abstract

End-to-end speech-to-text translation can provide a simpler and smaller system but is facing the challenge of data scarcity. Pre-training methods can leverage unlabeled data and have been shown to be effective on data-scarce settings. In this work, we explore whether self-supervised pre-trained speech representations can benefit the speech translation task in both high- and low-resource settings, whether they can transfer well to other languages, and whether they can be effectively combined with other common methods that help improve low-resource end-to-end speech translation such as using a pre-trained high-resource speech recognition system. We demonstrate that self-supervised pre-trained features can consistently improve the translation performance, and cross-lingual transfer allows to extend to a variety of languages without or with little tuning.

Index Terms: speech recognition, speech translation, pre-training, self-supervised learning, low-resource

1. Introduction

Recently, there has been much interest in end-to-end speech translation (ST) models [1, 2, 3, 4, 5, 6, 7], which, compared to traditional cascaded models, are simpler and computationally more efficient, can preserve more acoustic information and can avoid propagating errors from the speech recognition component. Large amounts of annotated data are usually required for achieving a good performance for such systems, but supervised training data for ST remain very limited.

On the other hand, unlabeled data are more accessible. Self-supervised techniques can exploit unlabeled data by learning a representation through, for instance, partial prediction or contrastive methods, and they have been shown effective for natural language [8, 9, 10] and speech processing [11, 12, 13]. In the latter case, several investigations on unsupervised or self-supervised pre-training have been conducted and applied to English automatic speech recognition (ASR) [12, 13], to multilingual ASR by training multilingual features [14] or transferring contrastive predictive coding (CPC) features to other languages [15].

In this paper, we are interested in whether self-supervised speech pre-training can effectively help speech-to-text translation on both high-resource and low-resource settings. In particular, we focus on the method of wav2vec [12] which makes use of contrastive predictive coding (CPC), the vector-quantized representation vq-wav2vec [13] and BERT features learned on top of the discretized representations [13].

We use speech features pre-trained on English, and first examine a high-resource within-language English-to-X ST setting (X denotes a non-English language), then we transfer the representations to 11 lower-resource X-to-English ST tasks. Transferring the parameters learned on a higher-resource ASR task has been shown to be an effective way to improve the performance and ameliorate the training of low-resource ST [16, 17, 18], thus we also study the interactions with self-supervised representations and whether we can effectively combine both methods.

We first demonstrate that compared to commonly used log-mel filterbank features, self-supervised features pre-trained on English can help improve English-to-X ST, but also transfer well to other languages even without requiring additional tuning. However, in the cross-lingual case, training data quantity and linguistic similarity may affect this gain. Further study shows that either fine-tuning the pre-trained input features or using a multilingual ASR model to fine-tune the final ST system can both improve the X-to-English ST. Finally, we show that when using an ASR model to pre-train ST systems, under certain training conditions, the ASR performance may not be a good indicator of the ST performance.

2. Methods

2.1. Self-supervised Learning for Speech Representations

Self-supervised learning allows to learn representations [8, 19, 11, 20, 21] through proxy tasks by, for instance, predicting some masked parts of the input, predicting future time-steps, contrasting with negative samples, or generating contextual data. In our case, we focus on three speech feature pre-training techniques which either makes use of CPC or a masked language model.

In this work, we explore **four** self-supervised approaches for learning speech representations in ST. The first and simplest representation is *wav2vec*[12], which learns speech representations through a future sample prediction task by optimizing a contrastive loss. The model consists of two convolutional neural networks, with an encoder network that takes raw audio as inputs and outputs a low-frequency representation to an aggregator, that creates a contextualized vector representation by combining the latent representation from multiple time steps. As a follow-up, *vq-wav2vec*[13] has an architecture similar to *wav2vec*, but with an additional quantization module between the encoder network and the aggregator, which discretizes the encoder's outputs before feeding them to the aggregator network. The output representation, as discrete tokens, can be consumed by natural language processing algorithms/models such as BERT from which we can extract representations for speech tasks. We also investigate an approach leveraging the pre-trained BERT, described in subsection 2.2.

2.2. Network architecture

For both ST and ASR tasks, our experiments are performed with a sequence-to-sequence BiLSTM attention-based encoder-decoder architecture following [4], but with a 3-layers decoder. Speech features are given as inputs to two non-linear $(tanh)$ layers, then passed to a stack of two convolutional layers. The output tensor is flattened and fed into three stacked bidirectional LSTM layers. The decoder is composed of two LSTM layers which output to a linear projection layer.

Table 1: *AST training data statistics. We also use the source language transcripts as the training data for ASR (if used).*

Pairs	Hours	Data	Pairs	Hours	Data
Fr-En	87h	CoVoST	Fa-En	20h	CoVoST
De-En	71h	CoVoST	Sv-En	1h	CoVoST
Es-En	21h	CoVoST	Mn-En	3h	CoVoST
Nl-En	4h	CoVoST	Zh-En	4h	CoVoST
Ru-En	10h	CoVoST			
It-En	13h	CoVoST	En-Fr	492h	MuST-C
Tr-En	3h	CoVoST	En-Ro	432h	MuST-C

For low-resource ST settings, we also investigate a hybrid BERT-backbone architecture, where we reuse the BERT model pre-trained on discretized speech features as the encoder. For the decoder, we keep the same architecture than the BiLSTM. While BERT is commonly used on monolingual tasks since it has been developed at first for natural language understanding, this allows to reuse it for a different goal and avoiding training an important number of parameters from scratch.

3. Experiments

3.1. Datasets

For English-to-X ST, we use the MuST-C [22] dataset, a corpus with audio recordings from English TED talks translated into 8 languages. The corpus comprises sentence-level aligned transcriptions and translations.

For X-to-English ST, we use the multilingual ST dataset CoVoST [23] from 11 languages (French, German, Dutch, Russian, Spanish, Italian, Turkish, Persian, Swedish, Mongolian and Chinese) to English, containing crowd-sourced speech with diverse speakers and accents on a variety of topics, from dialogue to movie scripts. For ASR, we use the English data from the corresponding Common Voice dataset (2019-06-12 release), with approximately 120 hours [24]. For the test set, we use the CoVoST test set for all the languages, and on the Tatoeba test set whenever it is available (i.e. for Fr, De, Nl, Ru and Es-En ST). Dataset statistics can be found in Table 1.

3.2. Self-supervised Pre-trained Models

In our experiments, we use the officially open-sourced wav2vec [12], vq-wav2vec (k-means) [13] and BERT models [13] [1] trained on the full 960h of Librispeech corpora [25].

3.3. Experimental Setups

3.3.1. Pipelines

For both high-resource and low-resource ST settings, we compute the log-mel filterbank features and extract the frozen learned features for direct ST training. For low-resource ST, we additionally pre-train an English ASR model with the corresponding speech features, then transfer the encoder or both the encoder and decoder parameters for warming-up ST training.

3.3.2. Preprocessing

For the preparation of transcript and translation, we normalize the punctuation, tokenize the text with sacreMoses and lowercase to align with previous settings [23] [22]. We remove the punctuations only from the transcripts. On CoVoST, we use a character-level vocabulary, with 54 characters including English alphabet and numerical characters, punctuations and the markers for fairseq [26] dictionary. On MuST-C, we choose a unigram vocabulary of size 10000 as in [27] to better balance the training time, as the sentences in MuST-C are generally longer. The vocabulary is obtained using SentencePiece [28].

We convert the raw MP3 files of Common Voice and Tatoeba into monochannel WAV format with a sampling rate of 16000 Hz. We then extract 80-dimensional log-mel filterbank features, using a 25ms window size and 10ms window shift. The dimension of the feature has been chosen as the best performing one among several tested. For pre-trained speech features, we use the features extracted respectively from a wav2vec model, a vq-wav2vec (kmeans) model pre-trained on Librispeech, and a BERT model pre-trained on Librispeech quantized with the corresponding vq-wav2vec model. Details of the models are provided in section 2. In the training set, samples with more than 3000 frames or having more than 400 characters are removed for GPU memory efficiency, and samples with less than 5 frames or 1 character are also removed to avoid non-significant or empty inputs.

3.3.3. Training and Inference

Training and inference use the fairseq framework [26]. We train using the Adam optimizer [29] with a learning rate of 1e-03 for BiLSTM models, and of 5e-05 for BERT-backbone models. We use a fixed learning schedule for BiLSTM models and a polynomial decay learning schedule for BERT-backbone models. In addition, we use SpecAugment [30] for both ASR and ST with LD policy but without time warping. When training with learned features, we change the policy along the frequency dimension proportional to the embedding size. It can be thought as a kind of dropout applied to the input.

At inference time, we use beam search with a beam size of 5. We evaluate using the last 5 checkpoints averaged. For ASR, the reported word error rate (WER) has been obtained using VizSeq [31]. For ST, the BLEU score [32] reported is case-insensitive and tokenized, obtained using sacreBLEU [33].

4. Results

4.1. English-to-X Speech Translation

In this experiment, we compare the baseline log-mel filterbank features (noted as fbank) with wav2vec, vq-wav2vec and BERT features on within-language English-to-X translation, where the source audio matches the language (English) on which the learned features have been pre-trained on.

Table 2 summarizes the results obtained using different input features with the BiLSTM architecture, on the MuST-C dataset, for the English-French and English-Romanian language pairs. We can see that for both pairs, pre-trained features outperform the baseline log-mel filterbank feature. The largest improvements are obtained using the wav2vec features, with respectively 2 and 1.1 BLEU gains. Note that the MuST-C dataset is composed of TED talks (spoken English), while pre-trained features were learned on Librispeech, without need for domain adaptation. Models using pre-trained features are also found to converge faster (Figure 1).

[1] These models are available for download at `https://github.com/pytorch/fairseq/tree/master/examples/wav2vec`

Table 2: *Results on the task of AST for MuST-C. The scores are computed in BLEU, on the tst-COMMON test set.*

	En-Fr	En-Ro
Di Gangi et al. [22]	22.3	13.4
Di Gangi et al. [6]	27.9	16.8
log-mel filterbank	27.8	17.1
wav2vec	**29.8**	**18.2**
vq-wav2vec	28.6	17.4
+ BERT base	28.6	17.3

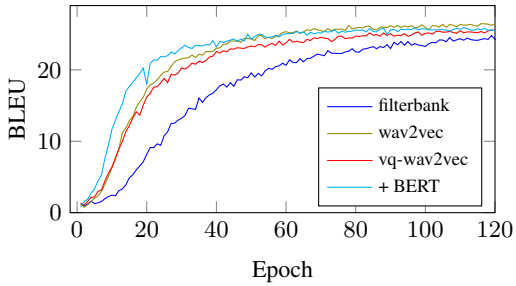

Figure 1: *Evolution of the BLEU score across epochs for different speech features on the MuST-C En-Fr dev set. The actual training has been performed until full convergence for all features.*

4.2. X-to-English Speech Translation

We now investigate whether pre-trained English speech features can be transferred to other languages for the X-to-En ST task.

4.2.1. Main Results

We investigate the low-resource X-to-English ST task. We consider both ST training from scratch and using an En ASR model to pre-train the ST components on the CoVoST dataset.

We report the ASR and ST results in Table 3. First, we find that while the pre-trained features are not helpful in very-low resource conditions, when there is a good baseline (either with a certain amount of data or combining with the ASR pre-training technique), they can consistently improve over the log-mel filterbank features and transfer well to other languages. On Fr-En ST, without any ASR pre-training, wav2vec features brought an improvement of 4.28/6.37 BLEU on CoVoST/Tatoeba. Second, the gain is cumulative with the ASR pre-training method to help improve low-resource ST performance, for all self-supervised features and almost all language pairs, except for Mongolian on which the systems failed to learn. Also, we observe that while on the ASR task, the most effective pre-trained feature is BERT, in the majority of X-to-En ST tasks, BERT features are outperformed by wav2vec or vq-wav2vec.

We plot Fr-En and Zh-En results in Figure 2 and Figure 3 for better visualization (the general trend for most other languages is similar to French). We observe that for French, wav2vec features are consistently outperforming the baseline. In the case of Chinese, log-mel filterbank is slightly worse when we directly train the ST, but outperforms learned representations when combining with ASR pre-training.

We also compare the results obtained on the BERT-backbone architecture with the baseline and other self-supervised approaches, on 5 languages pairs in Table 3. The

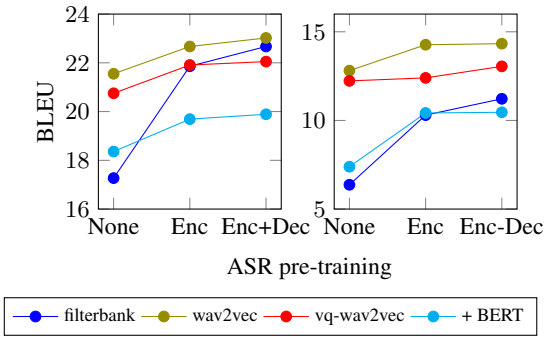

Figure 2: *Comparison of BLEU scores for Fr-En ST, with/without ASR pre-training, on CoVoST test set (left) and Tatoeba test set (right)*

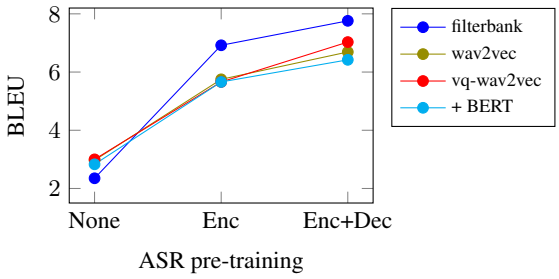

Figure 3: *Comparison of BLEU scores for Zh-En ST, with/without ASR pre-training, on CoVoST test set (*results averaged over 4 random seeds)*

parameters transferred from the pre-trained BERT encoder can lead to better performance on 4 language pairs compared to the systems trained from scratch, but it is not as effective as using ASR pre-training. What is surprising is that the encoder contains 123.6M parameters and can still be trained effectively on low-resource setting (ex. there are only 4h of training data for Dutch).

4.2.2. Transferring Features of Language X to English

We now study the impact of transferring features of language X to the pre-trained speech representations or systems. We first consider directly fine-tuning a pre-trained representation. Secondly, we consider training an ASR with both source (X) and target (English) data which will then be used to warm-up the ST training.

In the first approach, we compare frozen BERT features to the features fine-tuned on Common Voice speech data (2019-06-12 release) on Fr-En and Zh-EN ST tasks. The advantage of this approach is that no labeled data is required. Table 4 shows that fine-tuning is helpful in all cases, except for Zh-En ST without ASR pre-training. On both language pairs, combining fine-tuned features with ASR pre-training is more helpful when pre-training only the encoder.

In the second approach, we leverage ASR data and investigate the impact of mixing source language X with English data to train the ASR model which will then be used to fine-tune the encoder of the ST model. For both English and X, we use the Common Voice ASR training data. Table 5 shows the results for 4 language pairs from higher-resource to low-resource settings. While combining different languages may increase the WER of

Table 3: *Comparison of different speech features for English ASR and X-to-En AST. The first column indicates the WER of EN ASR models used to pre-train the ST. The ST results are on CoVoST/Tatoeba test set (when available). The ST languages are: German (De), French (Fr), Spanish (Es), Dutch (Nl), Russian (Ru), Italian (It), Turkish (Tr), Persian (Fa), Swedish (Sv), Mongolian (Mn) and Chinese (Zh). The baseline [23] is comparable to the case with ASR encoder pre-training, using log-mel filterbank features.*

Language Hours (test)	En	De 168.3	Fr 46.3	Es 3.5	Nl 8.2	Ru 8.2	It 12.8	Tr 3.8	Fa 23.9	Sv 1.0	Mn 2.9	Zh 3.7
Wang et al. [23]	-	7.6/7.5	21.4/10.9	6.1/1.9	3.4/5.0	4.8/1.1	6.5	3.1	2.8	1.9	0.3	5.6
fbank		3.1/1.5	17.3/6.4	0.8/0.5	0.1/0.1	1.3/0.1	0.5	1.1	0.3	0.2	0.4	2.4
wav2vec		6/**5.0**	**21.6/12.8**	0.4/0.4	0.3/0.5	**2.0**/0.1	0.4	0.9	1.6	0.3	0.2	**3.5**
vq-wav2vec		**6.1**/5.0	20.8/12.2	0.7/0.3	0.2/0.4	2.0/0.1	0.5	0.9	1.2	0.4	0.3	3
BERT-feature		2.8/1.2	18.4/7.4	0.2/0.2	0.1/0.2	1.4/0.1	0.4	0.6	0.2	0.3	0.1	2.8
BERT-backbone		6.7	16.4	3.4	2.1	5.1						
With ASR encoder pre-training												
fbank	34.3	7.2/6.6	21.9/10.3	5.5/1.9	3.3/3.9	5.1/0.8	7.0	**3.4**	2.7	1.8	0.2	**6.9**
wav2vec	32.6	**8.6/9.7**	**22.7/14.3**	**6.5/2.4**	3.8/5.0	**6.1/1.3**	**8.2**	3.4	3.2	**1.9**	0.1	5.8
vq-wav2vec	35	8.5/9.8	21.9/12.4	**6.5/2.4**	3.7/**5.4**	5.7/1.3	7.8	3.1	**3.3**	1.8	0.3	5.7
BERT-feature	32.1	7.6/8.3	19.7/10.4	5.7/2.4	**4.2**/4.2	5.7/1.0	6.6	3.0	3.1	1.8	0.3	5.7
With ASR encoder+decoder pre-training												
fbank		8.3/7.4	22.5/11.2	6.8/2.2	4.0/5.5	8.3/1.4	8.8	3.2	3.1	3.0	0.2	**8.2**
wav2vec		**9.7**/10.1	**23.0/14.3**	**7.2/3.6**	4.9/6.9	8.8/**1.8**	**9.7**	**3.4**	**3.7**	**3.7**	0.2	6.8
vq-wav2vec		9.6/**11.2**	22.1/13.1	6.9/3.3	**5.0/7.0**	**9.2**/1.7	9.0	3.3	**3.7**	3.2	0.3	7.0
BERT-feature		8.5/9.4	19.9/10.5	6.2/3.2	4.3/5.8	8.3/1.3	7.7	2.8	3.3	2.9	0.3	6.4

Table 4: *BLEU scores using BERT features fine-tuned on language X. The difference compared to the frozen features (row* BERT-feature *in Table 3) is in parentheses.*

ASR pre-training	Fr	Zh
None	18.7 (+0.4)	2.0 (-0.8)
Encoder	21.0 (+1.3)	6.8 (+1.1)
Encoder+Decoder	20.9 (+1.0)	6.7 (+0.3)

the ASR, it can still help improve the performance of the resulting ST in all cases. Also, for most languages, pre-trained representations can also improve over the baseline log-mel filterbank in this setting.

We observe that on Fr-En and Es-En ST, for all the 4 features, pre-training only the ST encoder with the En+X ASR is performing even better than pre-training both ST encoder and decoder with the En ASR (in Table 3). The largest gaps have been observed on BERT features, with respectively a difference of 1 and 1.6 BLEU for Fr-En and Es-En.

4.2.3. Influence of ASR Performance

The experiments in sections 4.2.1 and 4.2.2 suggest that when the training conditions differ, i.e. when comparing ASR models pre-trained on different features and/or on different languages, the ASR WER may not necessarily be correlated with the performance of the final AST.

Table 3 (column En) shows that while vq-wav2vec led to the worst performance on En ASR, in most cases, the final ST results are better than the systems pre-trained on En ASR with BERT features, whose WER is 2.9 points lower.

This effect is even more pronounced in Table 5, where in most cases, ASR models with higher WER can still help improve the translation performances.

Table 5: *WER for En+X ASR and BLEU for the corresponding ST, using encoder pre-training. Difference with respect to En ASR is in parentheses: for ASR, it is computed against the 1st column of Table 3, for AST against the respective languages of Table 3 for the encoder pre-training case. A, B, C and D refer to fbank, wav2vec, vq-wav2vec and BERT features, respectively.*

	De	Fr	Es	Zh
ASR				
A	35.9 (+1.6)	34.7 (+0.4)	34.7 (+0.4)	37.2 (+2.9)
B	33.5 (+0.9)	32.1 (-0.5)	32.9 (+0.3)	36.0 (+3.4)
C	35.4 (+0.4)	34.7 (-0.3)	35.9 (+0.9)	37.7 (+2.7)
D	35.0 (+2.9)	32.7 (+0.6)	32.8 (+0.7)	33.2 (+1.1)
AST				
A	8.3 (+1.1)	23.2 (+1.3)	7.4 (+1.9)	7.5 (+0.6)
B	9.3 (+0.7)	23.9 (+1.2)	8.4 (+1.9)	7.3 (+1.5)
C	9.5 (+1.0)	22.8 (+0.9)	7.7 (+1.2)	7.2 (+1.5)
D	8.4 (+0.8)	20.9 (+1.2)	7.8 (+2.0)	7.2 (+1.5)

5. Conclusion

We have shown that self-supervised representations can benefit the ST task. The resulting features can be directly transferred to other languages, and can be effectively combined with ASR pre-training for low-resource conditions to boost the performance. To improve the cross-lingual transfer on a given language, an effective way is to leverage ASR data by transferring the parameters learned on an ASR pre-trained on both higher-resource English and X data, or fine-tuning the pre-trained features on language X in an unsupervised way. Further work can include analyzing investigating the robustness of pre-trained features in other data conditions, and exploring multilingual settings.

6. References

[1] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *arXiv preprint arXiv:1612.01744*, 2016.

[2] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 949–959.

[3] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[4] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6224–6228.

[5] L. C. Vila, C. Escolano, J. A. Fonollosa, and M. R. Costa-jussà, "End-to-end speech translation with the transformer." in *IberSPEECH*, 2018, pp. 60–63.

[6] M. A. Di Gangi, M. Negri, and M. Turchi, "Adapting transformer to end-to-end spoken language translation," in *INTERSPEECH 2019*. International Speech Communication Association (ISCA), 2019, pp. 1133–1137.

[7] H. Inaguma, K. Duh, T. Kawahara, and S. Watanabe, "Multilingual end-to-end speech translation," *arXiv preprint arXiv:1910.00254*, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[10] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven pretraining of self-attention networks," *arXiv preprint arXiv:1903.07785*, 2019.

[11] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[12] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[13] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[14] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. v. d. Oord, "Learning robust and multilingual speech representations," *arXiv preprint arXiv:2001.11128*, 2020.

[15] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *arXiv preprint arXiv:2002.02848*, 2020.

[16] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," *arXiv preprint arXiv:1809.01431*, 2018.

[17] M. C. Stoian, S. Bansal, and S. Goldwater, "Analyzing asr pretraining for low-resource speech-to-text translation," *arXiv preprint arXiv:1910.10762*, 2019.

[18] C. Wang, Y. Wu, S. Liu, Z. Yang, and M. Zhou, "Bridging the gap between pre-training and fine-tuning for end-to-end speech translation," *arXiv preprint arXiv:1909.07575*, 2019.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[20] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[21] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[22] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "Must-c: a multilingual speech translation corpus," in *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019, pp. 2012–2017.

[23] C. Wang, J. Pino, A. Wu, and J. Gu, "Covost: A diverse multilingual speech-to-text translation corpus," *arXiv preprint arXiv:2002.01320*, 2020.

[24] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[26] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[27] A. D. McCarthy, L. Puzon, and J. Pino, "Skinaugment: Auto-encoding speaker conversions for automatic speech translation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7924–7928.

[28] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[31] C. Wang, A. Jain, D. Chen, and J. Gu, "Vizseq: A visual analysis toolkit for text generation tasks," *arXiv preprint arXiv:1909.05424*, 2019.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[33] M. Post, "A call for clarity in reporting bleu scores," *arXiv preprint arXiv:1804.08771*, 2018.