Investigating Self-supervised Pre-training for End-to-end Speech Translation

Ha Nguyen[1,2]*, Fethi Bougares*[3]*, Natalia Tomashenko*[2]*, Yannick Estève*[2]*, Laurent Besacier*[1]

[1]LIG - Université Grenoble Alpes, France
[2]LIA - Avignon Université, France
[3]LIUM - Le Mans Université, France

manh-ha.nguyen@univ-grenoble-alpes.fr

Abstract

Self-supervised learning from raw speech has been proven beneficial to improve automatic speech recognition (ASR). We investigate here its impact on end-to-end automatic speech translation (AST) performance. We use a contrastive predictive coding (CPC) model pre-trained from unlabeled speech as a feature extractor for a downstream AST task. We show that self-supervised pre-training is particularly efficient in low resource settings and that fine-tuning CPC models on the AST training data further improves performance. Even in higher resource settings, ensembling AST models trained with filter-bank and CPC representations leads to near state-of-the-art models without using any ASR pre-training. This might be particularly beneficial when one needs to develop a system that translates from speech in a language with poorly standardized orthography or even from speech in an unwritten language.

Index Terms: self-supervised learning from speech, automatic speech translation, end-to-end models, low resource settings.

1. Introduction

Self-supervised learning using huge unlabeled data has been explored with very promising results for image processing [1] and natural language processing [2]. Recent works investigated self-supervised representation learning from speech [3, 4, 5]. They were successful to improve performance on downstream tasks such as speech recognition. These recent works suggest that it is possible to reduce dependence on labeled data for building speech systems through acoustic representation learning. We investigate the possibility to leverage unlabeled speech for end-to-end automatic speech translation (AST). We focus on scenarios where (a) recordings in source language are not transcribed[1] (no ASR pre-training is possible), (b) only a small-medium amount of training data (speech aligned to translations) is available, (c) a larger amount of unlabeled speech can be used. This scenario is typical of situations when one builds a system that translates from speech in a language with poorly standardized orthography or even from an unwritten language.

In summary, our contributions are: (1) we propose an in-depth study on the impact of self-supervised pre-training for AST, (2) we show that fine-tuning pre-trained representations on the AST training data is beneficial and that self-supervised pre-training is particularly efficient in low resource settings, (3) even in high resource settings, ensembling models trained with filter-bank and self-supervised representations leads to near state-of-the-art models without using ASR pre-training, (4) we analyze the representations learnt and show that they allow to better discriminate phones, better align source and target sequences, and are more robust to speaker variability.

[1]Transcription not available or language poorly written.

2. Related Works

2.1. Self-supervised learning from speech

Self-supervised learning from speech consists in resolving pseudo-tasks not requiring human annotations as a pre-training to the real tasks to solve. These pseudo-tasks target predicting next samples or solving ordering problems. Autoregressive predictive coding (APC) [6, 7] considers the sequential structure of speech and predicts information about a future frame. An easier learning objective is introduced in Contrastive Predictive Coding (CPC) which consists in distinguishing a true future audio frame from negatives [3, 8, 9]. [5] shows that such representations are useful to improve several speech tasks while [4] extends those works by looking at the representations' robustness to domain and language shifts. In the same vein, [10] compares self-supervised and supervised pre-training for ASR and shows that CPC pre-training extracts features that transfer well to other languages, being on par or even outperforming supervised pre-training. Another promising way is to use speech enhancement as a task for feature representation learning [11, 12]. Finally, several self-supervised tasks can be jointly tackled to discover better speech representations [13].

2.2. End-to-end Automatic Speech Translation

Previous automatic speech-to-text translation (AST) systems operate in two steps: source language automatic speech recognition (ASR) and source-to-target text machine translation (MT). However, recent works have attempted to build end-to-end AST without using source language transcription during learning or decoding [14, 15] or using it at training time only [16]. Recently several extensions of these pioneering works were introduced: low resource AST [17], unsupervised AST [18], end-to-end speech-to-speech translation (*Translatotron*) [19], multilingual AST [20]. Improvements of end-to-end AST were also proposed using weakly supervised data [21] or adding a second attention mechanism [22]. While supervised pre-training for AST was investigated (see for instance [16]), we are aware of a single research group [5, 7] that investigated self-supervised pre-training for AST. However their experiments were done in a high resource setting and AST (for which only marginal gains were displayed) was solely investigated among other tasks, without an in-depth analysis of the representations learnt.

3. Self-supervised Pre-training from Speech

3.1. Contrastive predictive coding model

We use the self-supervised pre-training model introduced in [8] (*wav2vec*) which is based on contrastive predictive coding. The model uses (1) an encoder network that converts the audio sig-

Table 1: *Statistics of different How2 data partitions.*

Partition	#segments	#hours	#src words	#tgt words
10%	17,751	28	313K	295K
20%	35,858	56	626K	591K
30%	53,698	84	887K	940K
60%	107,676	169	1778K	1883K
full	179,438	281	2963K	3139K

nal into a latent representation (from raw speech samples x into a feature representation z), and (2) a context network that aggregates multiple time steps to build contextualized representations (from a sequence $z_{i-v}, ..., z_i$ into a context vector c_i).[2] The full model (encoder+context) is trained end-to-end to distinguish a sample z_{i+k} that is k steps in the future from negative samples $\tilde{z}$ uniformly chosen from the same audio sequence. A contrastive loss is minimized for each step $k = 1, ..., K$ and the overall loss is summed over different step sizes (more details in [8]).

3.2. Pre-trained models for English

We use an off-the-shelf model provided for English.[3] It is trained on *Librispeech* corpus [23]. We also investigate if fine-tuning the model on our task specific data is beneficial. For this, we fine-tune *wav2vec* on the full speech corpora used for our AST experiments (see next section). It is important to note that no transcripts nor translations are needed for this step which requires only raw speech. After fine-tuning *wav2vec*, we input the representations produced by the context network c_i to the AST encoder instead of filter-bank features (see Figure 1).

4. End-to-end Speech Translation Experiments

4.1. Experimental setup

4.1.1. Data

How2 corpus [24] is used for our main experiments. This corpus contains about 297.6 hours of speech, which is transcribed and translated into 3.3 million of English words and 3.1 million of Portuguese words respectively.[4] From this version of data, we first filter out too long sentences (sentences longer than 30 seconds or 400 characters). Then, in order to simulate lower resource scenarios, we randomly split the corpus into four sub-corpora of roughly 10%, 20%, 30%, and 60% of the filtered full corpus. Our splits guarantee that smaller partitions are fully included in the bigger ones. The statistics of all the partitions and the filtered version of full corpora can be found in Table 1.

4.1.2. Speech features and data augmentation

As shown in Figure 1, we extract either *wav2vec* features or filter-bank+pitch features (later denoted as *fbanks*) from speech input.[5] Depending on the experiments, mean and variance normalization (*MVN*) is optionally applied to the generated features. For *wav2vec* feature extraction, we either use an off-the-shelf model trained on LibriSpeech [23] or a model fine-tuned on How2 training set. *MVN* parameters are estimated on the speech translation training set and then applied to all train/dev/test sets. Overall, we have 4 different self-supervised representations named *wav2vec*, *wav2vec + norm*, *wav2vec + FT* (fined-tuned *wav2vec*) and *wav2vec + FT + norm*. All those *wav2vec* features are of dimension 512. We compare the above representations to conventional *filter-bank* features. Similar to [25], we extract 80-dimensional Mel filter-bank features, concatenated with 3-dimensional pitch features from windows of $25ms$, and a frame shift of $10ms$. *MVN* is used in the same manner as for *wav2vec* features. This gives us 2 additional speech representations named *fbanks* and *fbanks + norm* respectively (their dimension is 83).[6] Data augmentation through speed perturbation is also applied with factors of 0.9, 1.0, and 1.1 to the training data. We reuse the development set of our participation to the previous IWSLT2019 [25] ($1,984$ sentences randomly excluded from the training set). How2 val set is used as our test data. As for target text processing, we normalize punctuation marks, and tokenize the text into character-level using Moses script.[7]

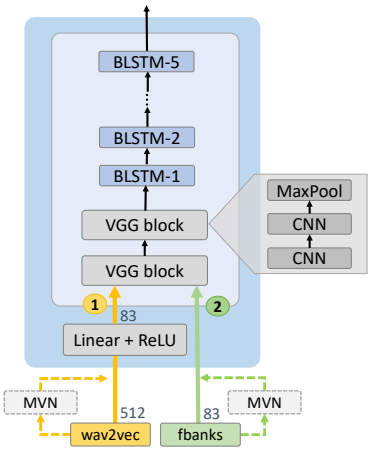


Figure 1: *Architecture of the speech encoder: a stack of two VGG blocks followed by 5 BLSTM layers. We use as input (1) wav2vec features (that pass through an additional projection layer to reduce their dimension from 512 to 83), or (2) filter-bank+pitch features. The input features are optionally normalized (MVN).*

4.2. Speech-to-text translation model

4.2.1. Architecture.

We use an attention-based encoder-decoder architecture, whose encoder is illustrated in Figure 1. The encoder is a stack of two VGG-like [26] CNN blocks followed by five 1024-dimensional BLSTM layers. Each VGG block contains two 2D-convolution layers just before a 2D-maxpooling layer, which aims to reduce both time (T) and frequency dimension (D) of the input speech features by a factor of 2. These two VGG blocks transform input speech features' shape from $(T \times D)$ to $(T/4 \times D/4)$. Bahdanau's attention mechanism [27] is used in all our experiments. The decoder is a stack of two 1024-dimensional LSTM layers. This model performed well at the IWSLT2019 E2E AST track [25], thus it is completely reused for all the experiments

[2] Practically each z_i encodes $30ms$ of speech every $10ms$. As for c_i, the total receptive field of the context network is $210ms$.

[3] https://github.com/pytorch/fairseq/blob/master/examples/wav2vec/

[4] Note that these statistics were measured on our version of How2 downloaded on July 12, 2019 [25].

[5] Our preliminary experiments on How2 10% with MFCC features which lead to similar performance as filter-bank are not presented here.

[6] For the rest of the paper *fbanks* will actually mean filter-bank+pitch

[7] https://github.com/moses-smt/mosesdecoder.

Table 2: *Detokenized case-sensitive BLEU scores measured on How2 val set of different models trained on different partitions of How2 corpus (EN-PT) with different speech features.* ***FT*** *means fine-tuned and* ***norm*** *stands for MVN normalization.*

No.	Feature	10% (28h)	20% (56h)	30% (84h)	60% (169h)	100% (281h)
1	wav2vec	11.33	26.75	30.83	36.33	41.02
2	wav2vec + FT	12.52	27.30	32.11	37.78	42.32
3	wav2vec + norm	16.52	27.33	31.27	37.62	41.08
4	wav2vec + FT + norm	18.50	27.68	32.17	37.75	41.30
5	fbanks	1.03	18.61	27.32	37.23	41.63
6	fbanks + norm	2.11	24.58	30.21	37.56	42.51
7	Ensemble [5, 6]		25.28	31.90	40.39	44.35
8	Ensemble [4, 6]		29.87	34.67	41.22	45.02
9	**Ensemble [1,2,3,4,5,6]**		**31.88**	**36.80**	**42.62**	**46.16**

with *fbanks* features presented throughout this paper. However *wav2vec* features have higher dimension (512) than *fbanks* (83). In order to compare both input representations with a similar parameter budget in the architecture (and also because training an architecture with input features of dimension 512 would be more computationally expensive), we add a projection block at the bottom of the encoder.[8] This block (containing a linear layer followed by a ReLU) reduces the *wav2vec*'s feature size from 512 to 83 (see Figure 1).

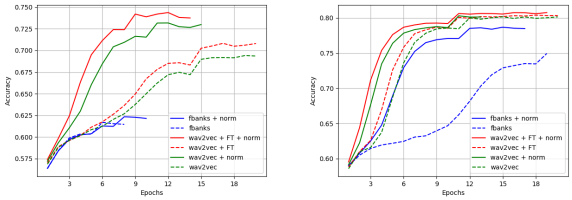
4.2.2. *Hyperparameters' details*

Models are trained in maximum 20 epochs with early stopping after 3 epochs if the accuracy on the dev set does not improve. Adadelta is chosen as optimizer and dropout is set to 0.3 on the encoder side. We decode all our models with beam size of 10.

4.3. Experimental results on How2

On each partition of How2 corpus, we train 6 models which take as input different speech representations presented in section 4.1.2, thus in total 30 models shown in Table 2. We evaluate on How2 val set, which contains $2,022$ segments (about 3.2 hours of speech), in the same conditions as our participation to IWSLT 2019 shared task. It is clear from the table that in low resource settings (28 and 56 hours), self-supervised representations (*wav2vec*) significantly outperform *fbanks*. Figure 2a confirms this and shows that models trained with *wav2vec* representations converge better and faster. The impact of normalization and fine-tuning is also notable from both Table 2 and Figure 2a. In very low resource settings (like 28 hours), fine-tuning *wav2vec* can greatly help, and with normalization, the performance further improves. In higher resource settings (169 and 281 hours of translated speech), differences between *wav2vec* and *fbanks* fade away (and so does the impact of fine-tuning and normalization). However, our ensembling experiments of lines 7 and 8 on 100% of How2 show that it is beneficial to ensemble the best system (*fbanks+norm*, line 6) with a system trained with *wav2vec* (*wav2vec+FT+norm*, line 4) rather than a better model (*fbanks*, line 5) also based on *filter-bank* features, even though *wav2vec+FT+norm* underperforms *fbanks* on this partition. Ensembling all our models (line 9) leads to $BLEU > 30$ even in very low resource training conditions (56 hours). Finally, in order to compare ourselves with the state-of-the-art [28], we decode How2 dev5 (a.k.a How2 test), which

[8]Our implementation of the *wav2vec* speech encoder, as well as the detailed recipes for our experiments can be found online: `https://github.com/mhn226/espnet/tree/interspeech2020`.



(a) *How2 10% (28 hours)* (b) *How2 20% (56 hours)*

Figure 2: *Learning curves (accuracy) of models trained on different partitions of How2.*

consists of $2,305$ segments (about 3.7 hours of speech), using the ensemble of all our models trained on the full corpus (line 9). This gives us near state-of-the-art BLEU: we obtain 46.16 on How2 val and 47.17 on How2 dev5. This latter score on dev5 is to be compared with 48.04 reported with an ensemble model in [28] where ASR and MT pre-training were used, as well as data augmentation with *SpecAugment*.

4.4. Validation on two other language pairs

To validate our results in low resource settings (56 and 84 hours), we train our models on two subsets of MuST-C [20] English-to-German and English-to-French training data (56 and 84 hours each, a training size similar to How2 20% and 30%). As illustrated by Table 3, MuST-C is more challenging than How2 (as confirmed by official IWSLT 2019 evaluation results [29]), but for both language pairs, *wav2vec* significantly outperform *fbanks*. This confirms that self-supervised pre-training is useful in low resource scenarios.

5. Analysis of Learnt Representations

This section tries to answer the question why *wav2vec* representation performs better than *filter-bank* features . The following subsections present the experiments which show that *wav2vec* might be (1) better at discriminating phones, (2) better at aligning source and target sequences, and (3) more robust to speaker variability.

5.1. Better phone discrimination

We first replicate an experiment from [8] for phoneme recognition on TIMIT [30]. Speech representations are extracted from train, dev and test split of TIMIT. A simple attentional encoder-decoder model is used: encoder with 4 BLSTM layers of hid-

Table 3: *AST BLEU on MuST-C for EN-DE and EN-FR.*

(a) *MuST-C 56 hours.*

Lang	Features	tst-COMMON	tst-HE
EN-DE	wav2vec	7.56	7.21
	wav2vec+norm	7.83	8.12
	fbanks	1.50	1.09
	fbanks+norm	4.89	4.87
EN-FR	wav2vec	12.08	12.41
	wav2vec+norm	12.58	12.58
	fbanks	0.54	0.00
	fbanks+norm	7.10	6.37

(b) *MuST-C 84 hours.*

Lang	Features	tst-COMMON	tst-HE
EN-DE	wav2vec	10.57	10.43
	wav2vec+norm	10.30	10.27
	fbanks	0.74	0.66
	fbanks+norm	7.68	7.84
EN-FR	wav2vec	16.18	16.68
	wav2vec+norm	16.84	16.37
	fbanks	1.65	0.97
	fbanks+norm	14.31	13.86

Table 4: *Phone error rate (PER %) on TIMIT dev and test set.*

No.	Feature	TIMIT dev	TMIT test
1	wav2vec	13.0	15.0
2	wav2vec + norm	13.9	15.8
3	fbanks	22.2	24.9
4	fbanks + norm	20.7	23.5

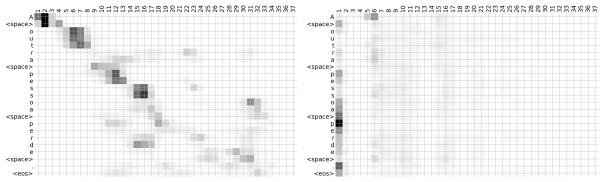
den size 320, decoder with 1 LSTM layer and location-based attention [31]. The results of Table 4 confirm that *wav2vec* representations (normalized or not) are much better at recognizing phones than *fbanks*.

5.2. Better source-target alignments

We evaluate the entropies of the soft alignments obtained with different speech representations in teacher forcing mode. Let α_{tj} be the alignment score between target token y_t and source speech frame x_j, we evaluate the entropy of the probability distribution α_t, $H_t = \sum_{j=1}^{|x|} \alpha_{tj} \log \alpha_{tj}$ for every target token. This measure is then averaged for all tokens at the corpus level (How 10%). A low entropy means the attention mechanism is confident in its source-target alignments (see example in Figure 3). Table 5 shows clearly that, in our low resource setting, *wav2vec* leads to better alignments (lower entropy) than *fbanks*. Fine-tuning and normalization of self-supervised representations also improve the soft alignments.

Table 5: *Averaged entropies of soft-alignments on How2 dev and val set. AST models trained on 10% partition of How2.*

No.	Feature	How2 dev	How2 val
1	wav2vec	0.66	0.66
2	wav2vec + FT	0.65	0.65
3	wav2vec + norm	0.57	0.57
4	wav2vec + FT + norm	0.51	0.51
5	fbanks	0.89	0.90
6	fbanks + norm	0.93	0.93



(a) *wav2vec* - $entropy = 0.67$ (b) *fbanks* - $entropy = 0.81$

Figure 3: *Soft alignments between source speech features and target text for sentence "A outra pessoa perde."*

5.3. Better robustness to speaker variability

Table 6: *Equal error rate (EER %) on the VoxCeleb1 test and LibriSpeech test sets for female (f) and male (m) speakers.*

No.	Feature	VoxCeleb	Libri (f)	Libri (m)
1	wav2vec	22.75	11.22	2.23
2	wav2vec + norm	20.93	10.54	1.79
3	fbanks	15.78	5.47	0.89
4	fbanks + norm	16.25	3.47	0.67

To investigate robustness to speaker variability, we trained several automatic speaker verification (ASV) systems using *wav2vec* or *fbanks* features. Models are trained on *LibriSpeech train-clean-360* dataset [23] using Kaldi [32]. ASV systems are based on x-vectors and probabilistic linear discriminant analysis (PLDA) [33]. To extract x-vectors, we used a time delay neural network (TDNN) model topology similar to the one described in [33]. Input features are *fbanks* or *wav2vec* (optionally normalized) while output corresponds to 921 speakers of the training corpus. ASV experiments are conducted on the *VoxCeleb1 test* [34] and *LibriSpeech test-clean* [23] sets.[9] ASV results (equal error rate - EER) are presented in Table 6. We observe that in all experiments, models trained on *wav2vec* features provide significantly higher EER in comparison with *fbanks*. This confirms our hypothesis that *wav2vec* representations remove speaker information from speech signal.[10]

6. Conclusion

We investigated the impact of self-supervised learning for end-to-end AST. It was shown that representations based on contrastive predicting coding (CPC) improve results significantly compared to baseline filter-bank, in low-medium resource conditions ($train < 100h$). Our explanation is that self-supervised representations show better phone discrimination, source-target alignments and speaker robustness.

7. Acknowledgements

This work was funded by the French Research Agency (ANR) through the ON-TRAC project under contract number ANR-18-CE23-0021.

[9]The trial and enrollment subsets of the *LibriSpeech test-clean* for the ASV task are described in more details in [35].

[10]We would also expect that mean and variance normalization increase EER but this is not the case. One explanation might be that normalization also removes channel variability and thus improves ASV.

8. References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[2] 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

[3] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," 2019.

[4] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," 2020.

[5] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," 2019.

[6] Y. Chung, W. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning," *CoRR*, vol. abs/1904.03240, 2019. [Online]. Available: http://arxiv.org/abs/1904.03240

[7] Y.-A. Chung and J. Glass, "Improved speech representations with multi-target autoregressive predictive coding," 2020.

[8] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1873

[9] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," 2019.

[10] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," 2020.

[11] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," 2020.

[12] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," 2020.

[13] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," 2019.

[14] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop on End-to-end Learning for Speech and Audio Processing*, 2016.

[15] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," in *Proc. of INTERSPEECH*, 2017.

[16] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," *CoRR*, vol. abs/1802.04200, 2018. [Online]. Available: http://arxiv.org/abs/1802.04200

[17] 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," *CoRR*, vol. abs/1809.01431, 2018. [Online]. Available: http://arxiv.org/abs/1809.01431

[18] Y. Chung, W. Weng, S. Tong, and J. Glass, "Towards unsupervised speech-to-text translation," *CoRR*, vol. abs/1811.01307, 2018. [Online]. Available: http://arxiv.org/abs/1811.01307

[19] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," *CoRR*, vol. abs/1904.06037, 2019. [Online]. Available: http://arxiv.org/abs/1904.06037

[20] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2012–2017. [Online]. Available: https://www.aclweb.org/anthology/N19-1202

[21] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," *CoRR*, vol. abs/1811.02050, 2018. [Online]. Available: http://arxiv.org/abs/1811.02050

[22] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," *CoRR*, vol. abs/1904.07209, 2019. [Online]. Available: http://arxiv.org/abs/1904.07209

[23] 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)*, 2015, pp. 5206–5210.

[24] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: a large-scale dataset for multimodal language understanding," in *ViGIL Workshop, NeurIPS*, 2018.

[25] H. Nguyen, N. Tomashenko, M. Z. Boito, A. Caubriere, F. Bougares, M. Rouvier, L. Besacier, and Y. Esteve, "ON-TRAC consortium end-to-end speech translation systems for the IWSLT 2019 shared task," in *Proc. of IWSLT*, 2019.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of ICLR*, 2015.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. of ICLR*, 2015.

[28] H. Inaguma, S. Kiyono, K. Duh, S. Karita, N. E. Y. Soplin, T. Hayashi, and S. Watanabe, "ESPnet-ST: All-in-one speech translation toolkit," *arXiv preprint arXiv:2004.10234*, 2020.

[29] J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M. Federico, "The iwslt 2019 evaluation campaign," in *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, 2019.

[30] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus cdrom," 1993.

[31] N.-Q. Luong, L. Besacier, and B. Lecouteux, "Towards accurate predictors of word quality for machine translation: Lessons learned on french - english and english - spanish systems," *Data and Knowledge Engineering*, 2015.

[32] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann *et al.*, "The Kaldi speech recognition toolkit," Tech. Rep., 2011.

[33] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[34] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.

[35] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé, and M. Todisco, "Introducing the VoicePrivacy initiative," in *Interspeech*, 2020.