Interspeech 2022
18-22 September 2022, Incheon, Korea

M-Adapter: Modality Adaptation for End-to-End Speech-to-Text Translation

Jinming Zhao, Hao Yang, Ehsan Shareghi, Gholamreza Haffari

Department of Data Science & AI, Monash University

first.last@monash.edu

Abstract

End-to-end speech-to-text translation models are often initialized with pre-trained speech encoder and pre-trained text decoder. This leads to a significant training gap between pre-training and fine-tuning, largely due to the modality differences between speech outputs from the encoder and text inputs to the decoder. In this work, we aim to bridge the modality gap between speech and text to improve translation quality. We propose M-Adapter, a novel Transformer-based module, to adapt speech representations to text. While shrinking the speech sequence, M-Adapter produces features desired for speech-to-text translation via modelling global and local dependencies of a speech sequence. Our experimental results show that our model outperforms a strong baseline by up to 1 BLEU score on the Must-C En→DE dataset.[1]

Index Terms: speech translation, modality adaptation

1. Introduction

Speech-to-text translation (ST) is the task of translating audio signals in one language to text in a foreign language. It has been conventionally approached with a cascaded architecture comprising automatic speech recognition (ASR) and machine translation (MT) components. The more recent end-to-end (E2E) approach has attracted great attentions, which involves an audio encoder taking audio signals as input and a text decoder producing a translated text. The E2E approach alleviates the issues of error propagation and high latency with in the cascaded methods [1, 2]. That said, the E2E approach often requires large amounts of paired audios and translated text, which is not available for most language pairs. A common practice to remedy the data scarcity issue is to pre-train the audio encoder and text decoder, and initialize the ST model with the pre-trained encoder and decoder [3, 4]. Successful developments in this direction were made by utilizing a pre-trained wav2vec 2 [5] as an audio encoder, and the decoder of mBart [6] as a text decoder in state-of-the-art (SOTA) ST systems [7, 8].

However, this method suffers from two inherent bottlenecks mainly due to the modality differences between speech and text: (i) Audio signals are several orders of magnitude longer than their transcripts, and they contain lots of redundancy [9], aggravating the alignment difficulty between speech outputs and text input, and (ii) compared to text, audio signals exhibit a much higher degree of variations caused by speaker and noise which amounts to learning difficulties. We frame both issues under *modality gap* between speech and text representations.

There are two lines of research to bridge the modality gap. On the one hand, prior works propose feature selection modules to compress speech length. [7, 10] use convolutional neural networks (CNNs) to collapse a fixed number of adjacent feature vectors into a single one. Yet, CNNs only model local information, thus exposing the risk of information loss [11]. [9, 12, 13] propose to select important features dynamically, but they require transcriptions or phonemes, which may not exist for all language pairs [2]. Alternatively, a text encoder can be attached to an audio encoder in a serial or parallel fashion [14, 15, 16] or be integrated with an acoustic encoder, forming a unified encoder [17, 4, 18, 19]. However, this approach is either coupled with a feature selection module, or requires large amounts of ASR and MT data, which makes training inefficient.

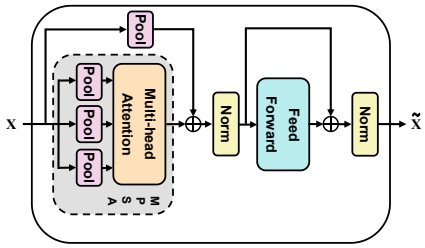

Figure 1: *Overview of M-Adapter.*

In this work, we aim to lift the modality gap between speech and text via reducing speech length and generating features more desirable for translation. Considering that Transformer [20] has achieved huge success in applications in various domains, we are motivated to unleash its potential further by adapting it to be a modality adapter. We propose M-Adapter, a Transformer-based architecture combined with convolutional layers, which models global and local dependencies of speech features while shrinking speech sequences. Figure 1 depicts M-Adapter. The key to M-Adapter is Multi-head Pooled Self-Attention (MPSA), inside which convolution layers pool Q, K, V matrices, linearly projected from input X, to intermediate matrices. An additional pooling module is applied to X. Together, they reduce the dimentionality of X as the output of the current layer, mitigating the aforementioned length mismatch issue. Moreover, Transformer exploits long-range global context, while convolution layer is good at modeling local features; our hypothesis is a combination of the two establishes global and local interactions, which is essential for producing higher-level features desired for ST.

Our main contributions are:

- This is the first work on utilizing a Transformer architecture to reduce length and adapt features of a speech sequence for ST. We highlight the importance of establishing global and local interactions via self-attention and convolutions, hoping to improve our understanding of feature generation for ST.
- Without using any additional data, our model outperforms strong baselines across 3 Must-C language pairs, with an average improvement of 0.78 BLEU scores.
- M-Adapter helps to reduce the performance gap in various resource conditions by a large margin.

[1] Our code is available at https://github.com/mingzi151/w2v2-st.

10.21437/Interspeech.2022-592

2. Preliminaries

2.1. Transformer

Transformer [20] is a highly modularized neural network, consisting of several Transformer blocks. Each block has two main modules, i.e., a multi-head self-attention (MSA) layer and a feedforward layer (FFN). MSA accepts an input sequence from the previous block and considers the relevance of features at other positions while it is being applied to a certain position. FFN performs non-linear transformation on these features to produce an output sequence for the next block. The two modules are wrapped by residual connection and layer normalization. Each Transformer block does not change the output sequence length. We refer readers to the original papers for details.

2.2. Pre-trained models

It is not always possible to train a ST model from scratch successfully, due to limited ST corpora and computation resources. Leveraging pre-trained models trained on massive data is a promising direction, as they serve as good initiation points. It is particularly beneficial for low-resource ST. Naturally, this requires a pre-trained acoustic encoder and a pre-trained text decoder.

Pre-trained speech encoders. Pre-trained models have been explored in the speech domain, to encode general-purpose knowledge. Typical methods include generative learning [21, 22, 23], multi-task learning [24] and discriminative learning [5, 25]. Theses acoustic models (most are Transformer-based) are trained with well-designed objective functions during the pre-training phase and used as a standalone acoustic encoder for downstream tasks.

Pre-trained text decoders. Pre-training a text decoder can be done independently (e.g., GPT2 [26]) or jointly with an encoder for sequence-to-sequence tasks (e.g., mT5 [27], mBart [6]). With the former approach, after the pre-training phase, the text decoder needs to be fused with additional encoders or adapters [28], which increases the complexity of architectures. For the latter approach, the decoder component can usually be used as an individual module without architectural modifications.

3. E2E ST with M-Adapter

Training a cascaded system with pre-trained models is a sub-optimal option. Not only do the common problems with cascaded approaches remain unsolved, but also it requires intermediate transcripts, which is, again, not always available. Therefore, in this work we investigate how to effectively leverage pre-trained models by addressing the modality mismatch between speech and text. More specifically, we will use wav2vec 2 (W2V2) as an speech encoder and mBart decoder as the text decoder, and integrate them with our proposed M-Adapter.

3.1. Pre-trained Modules

W2V2 is pre-trained on large untranscribed data and it can achieve excellent transcription performance through fine-tuning on a small amount of parallel ASR data. It is pre-trained with a contrastive objective that empowers the model to distinguish a true masked segment and those produced by the model. Its Transformer layers allow the model to encode contextual information surrounding the masked segment. After the training is complete, only its CNN feature extractor and Transformer layers are kept for downstream tasks.

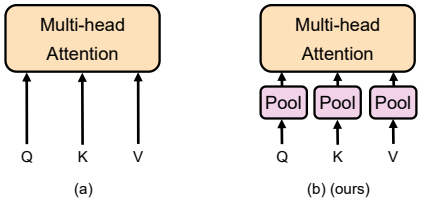

Figure 2: *Multi-head Self-Attention (left) vs Multi-head Pooled Self-Attention (right)*

mBart is a Transformer-based Seq2Seq denoising auto-encoder model, trained on massive amounts of monolingual and multilingual data. The training objective is to reconstruct an input sequence conditioned on a corrupted version. For our task, we use its decoder component.

3.2. M-Adapter

To adapt speech to text, we propose to replace MSA with MPSA, shown in Figure 2. In contrast to MSA which does not modify the sequence length, MPSA pools the sequence to reduce its length. M-Adapter aggregates local fine-grained features via pooling operation while capturing global information via MSA. It also encourages the model to capture better features for the task at hand via the interactions between MSA and convolutional layers.

Multi-head Pooled Self-Attention Formally, given an input sequence $\mathbf{X} \in \mathcal{R}^{L\times D}$ where L is the sequence length and D is the embedding dimension, same as MSA, MPSA first linearly projects $\mathbf{X}$ to query, key, and value matrices: $\mathbf{Q} \in \mathcal{R}^{L\times D}, \mathbf{K} \in \mathcal{R}^{L\times D}, \mathbf{V} \in \mathcal{R}^{L\times D}$.

Next, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are pooled with modules $\mathrm{Pool_Q}$, $\mathrm{Pool_K}$ and $\mathrm{Pool_V}$. Each pooling module consists of a 1D convolutional layer parameterized with kernel size k, stride s and padding p. Three new matrices are obtained:

$$Q' = \mathrm{Pool_Q}(Q) \qquad K' = \mathrm{Pool_K}(K) \qquad V' = \mathrm{Pool_V}(V)$$

where $\mathbf{Q}' \in \mathcal{R}^{L'\times D}$, $\mathbf{K}' \in \mathcal{R}^{L'\times D}$ and $\mathbf{V}' \in \mathcal{R}^{L'\times D}$ and L' is the length of the new sequence, which is determined by

$$L' = \left\lfloor \frac{L + 2p - k}{s} \right\rfloor + 1$$

Attention scores are then calculated based on the new matrices as follows.

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(Q'K'^T/\sqrt{d}\right)V'$$

where $\sqrt{d}$ normalizes the inner product matrix of $\mathbf{Q}'$ and $\mathbf{K}'$. Practically, we use h attention heads, each of which performs attention operation on a non-overlapping subset of d of the D dimensions of X. This yields h sequences which are then concatenated to $\mathbf{H} \in \mathcal{R}^{L'\times D}$.

Pooled Input We cannot add $\mathbf{X}$ and $\mathbf{H}$ directly (see Figure 1), as they now have different lengths. To overcome the issue, another 1D convolution pooling layer, $\mathrm{Pool_X}$, is used to convert $\mathbf{X}$ to $\mathbf{X}' \in \mathcal{R}^{L'\times D}$, to allow for subsequent addition and layer norm operations.

Each M-Adapter layer thus reduces the sequence length by a factor of s.

		Model	Encoder Size (M)	Compression	Extra Data	BLEU
Decoder Fine-tuned	Baselines	W2V2-cnn-mBart$^{\diamond}$	117	8:1	-	26.12
		W2V2-tran-mBart$^{\diamond}$	147	1:1	-	26.56
		XSTNet*	117	4:1	✓	25.5
		JT-S-MT^{+}	76	1:1	✓	26.8
	Ours	W2V2-mAda$_1$-mBart	154	8:1	-	27.00
		W2V2-mAda$_3$-mBart	185	8:1		**27.13**
Decoder Frozen	Baselines	W2V2-cnn-mBart$^{\diamond}$	319	8:1	-	26.45
		W2V2-tran-mBart$^{\diamond}$	349	1:1	-	26.91
	Ours	W2V2-mAda$_1$-mBart	357	8:1	-	27.60
		W2V2-mAda$_3$-mBart	386	8:1	-	**27.73**

Table 1: *BLEU results on Must-C EN→DE. Top rows: a decoder is fully (XSTNet and JT-S-MT) or partially (the rest) trained, while an encoder is being trained. Bottom rows: a decoder is frozen and W2V2 is fully trained.* ***Bold****: best results in each setting.* $^{\diamond}$*: replicated.* *, $^{+}$*: taken from [17], [4].*

Model	BLEU DE	RO	FR	△
Baseline				
W2V2-cnn-mBart	26.12	23.72	36.92	-
Proposed				
W2V2-mAda$_3$-mBart	27.13	24.62	37.34	0.78

Table 2: *BLEU for EN→DE, RO, and FR.*

4. Experiments

4.1. Data

We experiment with Must-C [29], a multilingual ST corpus built from Ted talks whose source language is English (EN). We focus on EN→DE (German), RO (Romanian) and FR (French). We follow the instructions in [7] to preprocess data. We develop our models on the dev set and report results on the tst-COMMON set. No transcripts or extra ASR/MT data are used.

4.2. Experimental Settings

4.2.1. Implementation details

W2V2 models are pre-trained on large unlabeled speech data and can be optionally fine-tuned on ASR data before used for downstream tasks. The model we use has a "large" architecture with the quantization module removed, pre-trained on 53.2k-hour of unlabelled data and fine-tuned on 960-hour of labelled data and pseudo-labels. The mBart model[2] we use has 12-layer Transformer blocks for the decoder, fine-tuned with the multilingual MT task (English→50 languages). Similar with [7], we employ a lightweight adapter (lit-adapter)[3] before M-Adapter to adapt pre-trained models to new tasks. We deploy N number of M-Adapter blocks. The M-Adapter modules lead to reduced length by stride factors of s^N. We implement two variants of our model: i) W2V2-mAda$_1$-mBart: it has 1 M-Adapter layer (short for mAda$_1$) with 8-stride 1D convolutional pooling modules and kernel size and padding of 8 and 4. ii) W2V2-mAda$_3$-mBart: this model has 3 M-Adapter layers (mAda$_3$) with kernel size, stride, padding as 3, 2 and 1. Both models shrink an input sequence by 8. We use BLEU[4] to measure translation quality.

4.2.2. Training strategy

To train our model, we follow the two-step LayerNorm and Attention training strategy (LNA-2step) used in [8]. In the first step, we train our M-Adapter layers, together with lit-adapter, and keep W2V2 and mBart frozen. In the second step, we fine-tune the adapters and a portion of W2V2 and mBart, including layer normalization, encoder self-attention and encoder-decoder attention. To further quantify the benefits that our approach bring to W2V2, we experiment with training the entire W2V2 while keeping the mBart frozen. Our models are trained for 32 epochs until early stopping is reached.

4.3. Baselines

We compare our model with the following baselines:

- **W2V2-cnn-mBart** [7]: It is an E2E ST models using pre-trained models with a similar architecture as ours, except that it uses 3 CNN layers as the length adapter. The same W2V2 and mBart checkpoints are employed. It has the same level of length compression as our model.
- **W2V2-tran-mBart**: M-Adapter layers are replaced with 3 Transformer (thus short for "tran") layers.
- **XSTNet** [17]: The encoder of the model consists of a wav2vec 2 (base) and convolution layers and one unified Transformer encoder to jointly encode speech and transcript.
- **JT-S-MT** [4]: It has a unified encoder (initialized with a pre-trained speech encoder and a text encoder) to perform ST and MT tasks. It leverages massive labelled MT data.

4.4. Main Results

Table 1 shows the results on Must-C EN→DE in two conditions: decoder fined-tuned and decoder frozen. In the top rows where the decoder is fine-tuned, our models surpass the baselines significantly. Particularly, our models surpass W2V2-cnn-mBart, the previous SOTA model on this dataset, by a large margin.[5] In the bottom rows where the decoder is fixed, while our models still outperform the baselines, our models improve BLEU scores compared to our own models from the top rows by an average of 0.60. This indicates the effectiveness of our method in unleashing the potential of W2V2. It also highlights the importance of the quality of speech representation for ST. Interestingly, with deeper layers, W2V2-mAd$_3$-mBart brings a marginal improvement compared to W2V2-mAd$_1$-mBart in both settings. We speculate the reason is that W2V2 produces high-level features [30] and a single M-Adapter layer is sufficient to transform these features to more complex features desired for translation.

[2]For the rest of our paper, we refer mBart's decoder as mBart.

[3]Its main components are two linear layers. We will drop the mention of this adapter since it is used in all experiments, including the baselines we replicated.

[4]https://github.com/mjpost/sacreBLEU

[5]It may appear our performance gain comes from more parameters, but we show that it is not the case in Subsection 5.1.

We also experiment on Must-C EN→RO, FR, with 3 M-Adapter layers attached and the decoder fine-tuned. Table 2 summarizes the results.[6] On average, our method improves translation quality by 0.78 in BLEU compared to the baseline.

5. Analysis

5.1. Impact of Length Reduction

Table 3 demonstrates the impact of the amount of length compression on the final results. To control our settings (e.g., model size), we experiment with W2V2-mAda$_1$-mBart, keep the kernel size constant, and vary stride from 2 to 8. We set the compression cap to 8, because beyond 8 the length of a shrunk sequence will be less than that of the corresponding text sequence. Given the module size, constant improvements in BLEU are observed as stride increases, implying a great deal of redundancy in speech representations and the necessity of compressing them.

Kernel	8	8	8	8
Stride	8	6	4	2
BLEU	27.00	26.79	26.51	26.10

Table 3: *Varied degree of compression for W2V2-mAd$_1$-mBart.*

5.2. Global and Local Interactions

We investigate why our method is superior than the baselines. We hypothesize that M-Adapter does more than length reduction; more fundamentally, it builds good global and local interactions, via a combination of Transformer and convolutional layers in that the former excels at modelling global context and the latter at capturing fine-grained, local features.

Positions of Pooling Modules: To validate our hypothesis, we conduct ablation study on W2V2-mAda$_3$-mBart by changing the positions of the pooling modules inside M-Adapter, i.e., pooling takes place (1) before MAS (b4p); (2) before feedforward (b4f); and 3) after the second layernorm (b4o). The BLEU scores for these variants are 26.27 (b4p), 25.38 (b4f) and 24.94 (b4o), compared to the current result (27.13). This suggests that local and global information are well blended at this position.

Removal of Global Capacity: To further investigate the global capacity of M-Adapter, we remove MPSA at the inference time and let M-Adapter perform local aggregation only. Our results show that the performance of W2V2-mAda$_3$-mBart drops from 27.13 to 26.84, highlighting the significance of considering long-range information in producing better speech features. MPSA, complemented by FFN and other components, performs better than bare CNN layers as in the baseline.

Perturbation on Representations: For the purpose of examining the quality of representations, we deliberately perturb speech features at the inference time at two levels, i.e., the output of W2V2 and the outputs of the adapters, at ratios of 10%, 20% and 50%. Table 4 compares the perturbation effects on W2V2-cnn-mBart and W2V2-mAd$_3$-mBart. On the one hand, perturbation at the W2V2-level has bigger impacts on the latter than the former. We conjecture that the reason is that perturbing vectors at this level breaks the global and local linkage established by M-Adapter. On the other hand, perturbation at the adapter-level has less influence on W2V2-mAd$_3$-mBart than W2V2-cnn-mBart. We believe this is because each vector produced by M-Adapter contains both local (with convolution layer) and global (with self-attention) information. When vectors are corrupted, the information carried in them can be inferred from the uncorrupted ones. Effectively, more information is preserved in remaining vectors with M-Adapter than CNN, thus making M-Adapter more robust.

[6] We expect to see the same trend as in Table 1 for EN → RO, FR.

Perturb.	%	Model: W2V2-cnn-mBart	Model: W2V2-mAd$_3$-mBart
none	-	26.12	27.13
W2V2	10	25.53 (2%)	26.46 (2%)
	20	24.27 (7%)	24.89 (8%)
	50	12.72 (51%)	9.69 (64%)
adapter	10	23.78 (9%)	25.25 (7%)
	20	20.08 (23%)	23.00 (15%)
	50	9.58(63%)	12.71 (53%)

Table 4: *BLEU scores with perturbation on representations at W2V2 and Adapter outputs, with different ratios. (↓) indicates performance drop compared to the original models.*

5.3. Low-Resource Settings

We examine the robustness of our model in different resource conditions. We conjecture that compressed, better quality speech representations bring more data efficiency. We stimulate high- (408 hours), mid- (204, 82 hours) and low-resources (41 hours) settings. Figure 3 shows that in high-, mid-resource conditions, W2V2-mAd$_3$-mBart has less degradation in BLEU, compared to W2V2-cnn-mBart, which has validated our hypothesis. In the low-resource setting, the two models perform similarly, which is expected, as M-Adapter does require sufficient amounts of training data.

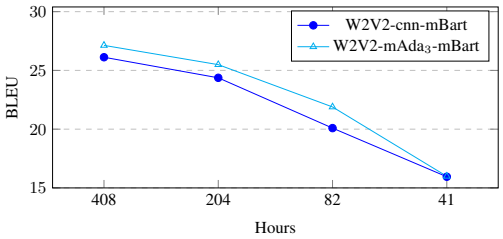

Figure 3: *Model performance in various resource conditions.*

5.4. Sparsity

We measure the sparsity of representations produced by above two models, with Hoyer metric [31]. In essence, Hoyer is a proportion of the L^2 over L^1 norm; higher Hoyer score indicates more sparsity, less utilization of the representation space. The average Hoyer scores for the two models are 0.7930 and 0.4489, respectively. This indicates that the representations produced by our model make better use of the space. We speculate this might serve as another indicator measuring an acoustic encoder. We will investigate further on its implications in our future work.

6. Conclusion

We propose M-Adapter, a novel Transformer-based architecture to lift the modality gap between speech and text representations. We demonstrate the importance of reducing speech length and that of establishing global and local interactions via a combination of Transformer and convolution layers in producing high-level speech features that are more suitable for ST. Our method surpasses strong baselines by a large margin.

7. References

[1] M. Sperber and M. Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7409–7421.

[2] J. Chen, M. Ma, R. Zheng, and L. Huang, "Specrec: An alternative solution for improving end-to-end speech-to-text translation via spectrogram reconstruction," *Proc. Interspeech 2021*, pp. 2232–2236, 2021.

[3] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *Proc. Interspeech 2017*, pp. 2625–2629, 2017.

[4] Y. Tang, J. Pino, X. Li, C. Wang, and D. Genzel, "Improving speech translation by understanding and learning from the auxiliary text translation task," *arXiv preprint arXiv:2107.05782*, 2021.

[5] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[6] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.

[7] G. I. Gállego, I. Tsiamas, C. Escolano, J. A. Fonollosa, and M. R. Costa-jussà, "End-to-end speech translation with pre-trained models and adapters: Upc at iwslt 2021," in *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, 2021, pp. 110–119.

[8] X. Li, C. Wang, Y. Tang, C. Tran, Y. Tang, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Multilingual speech translation from efficient finetuning of pretrained models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 827–838.

[9] B. Zhang, I. Titov, B. Haddow, and R. Sennrich, "Adaptive feature selection for end-to-end speech translation," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2533–2544.

[10] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. Pino, "Fairseq s2t: Fast speech-to-text modeling with fairseq," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, 2020, pp. 33–39.

[11] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[12] E. Salesky and A. W. Black, "Phone features improve speech translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2388–2397.

[13] M. Gaido, M. Cettolo, M. Negri, and M. Turchi, "Ctc-based compression for direct speech translation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 690–696.

[14] C. Xu, B. Hu, Y. Li, Y. Zhang, S. Huang, Q. Ju, T. Xiao, and J. Zhu, "Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 2619–2630.

[15] Y. Liu, J. Zhu, J. Zhang, and C. Zong, "Bridging the modality gap for speech-to-text translation," *arXiv preprint arXiv:2010.14920*, 2020.

[16] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9161–9168.

[17] R. Ye, M. Wang, and L. Li, "End-to-end speech translation via cross-modal progressive training," *Proc. Interspeech 2021*, pp. 2021–1065, 2021.

[18] R. Zheng, J. Chen, M. Ma, and L. Huang, "Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 736–12 746.

[19] C. Han, M. Wang, H. Ji, and L. Li, "Learning shared semantic space for speech-to-text translation," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2214–2225.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[22] Y.-C. Chen, S.-F. Huang, H.-y. Lee, Y.-H. Wang, and C.-H. Shen, "Audio word2vec: Sequence-to-sequence autoencoding for unsupervised learning of audio segmentation and representation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1481–1493, 2019.

[23] S. Ling and Y. Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[24] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

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

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[27] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498.

[28] Z. Sun, M. Wang, and L. Li, "Multilingual translation via grafting pre-trained language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2735–2747.

[29] R. Cattoni, M. A. Di Gangi, L. Bentivogli, M. Negri, and M. Turchi, "Must-c: A multilingual corpus for end-to-end speech translation," *Computer Speech & Language*, vol. 66, p. 101155, 2021.

[30] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *IEEE Automatic Speech Recognition and Understanding Workshop-ASRU 2021*, 2021.

[31] N. Hurley and S. Rickard, "Comparing measures of sparsity," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.