ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447926

CROSS-MODAL MULTI-TASKING FOR SPEECH-TO-TEXT TRANSLATION VIA HARD PARAMETER SHARING

Brian Yan[1], *Xuankai Chang*[1], *Antonios Anastasopoulos*[3], *Yuya Fujita*[4], *Shinji Watanabe*[1,2]

[1]Carnegie Mellon University, US, [2]Johns Hopkins University, US,
[3]George Mason University, US, [4]Yahoo Japan Corporation, JP

ABSTRACT

Recent works in end-to-end speech-to-text translation (ST) have proposed multi-tasking methods with *soft parameter sharing* which leverage machine translation (MT) data via secondary encoders that map text inputs to an eventual cross-modal representation. In this work, we instead propose a ST/MT multi-tasking framework with *hard parameter sharing* in which all model parameters are shared cross-modally. Our method reduces the speech-text modality gap via a pre-processing stage which converts speech and text inputs into two discrete token sequences of similar length – this allows models to indiscriminately process both modalities simply using a joint vocabulary. With experiments on MuST-C, we demonstrate that our multi-tasking framework improves attentional encoder-decoder, Connectionist Temporal Classification (CTC), transducer, and joint CTC/attention models by an average of +0.5 BLEU without any external MT data. Further, we show that this framework incorporates external MT data, yielding +0.8 BLEU, and also improves transfer learning from pre-trained textual models, yielding +1.8 BLEU.[1]

Index Terms— ST, MT, multi-tasking, transfer learning

1. INTRODUCTION

One of the preeminent challenges in end-to-end speech-to-text translation (ST) is that of data scarcity. There are relatively small amounts of labeled ST data compared to automatic speech recognition (ASR) and machine translation (MT) data, as well as unpaired speech and text data. Simply pseudo-labeling ASR data with strong MT models has proven to be effective [1, 2]; however, synthesizing speech for MT data using TTS has proven to be more complex and less effective [1, 3]. So what techniques aside from data augmentation can leverage textual data towards improving ST systems?

A popular answer amongst recent works is multi-tasked learning, where models are jointly optimized to perform MT and ST. Many proposed multi-tasking methods employ varying degrees of *soft parameter sharing* [4], where some parameters are shared while others are task-specific. Generally, these methods use modality-specific modules that map continuous speech inputs and discrete text inputs into an eventual common latent space [5–13]. In this work, we refer to this family of approaches as **soft multi-tasking** methods. These methods often require cross-modal regularization [8–12] to encourage greater similarity between speech and text representations, demonstrating that the gap speech and text modalities must be reduced to enable the benefits of soft multi-tasked learning.

An alternative approach to reducing the speech-text modality gap is to first convert continuous speech signals into discrete sequences. Recent works have shown that discrete speech sequences obtained by applying k-means clustering on self-supervised learning (SSL) representations contain sufficient semantic information to be effective system inputs for ASR [14] and ST [15]. Similarly, speech discretization has been applied in TTS [16–19] and speech-to-speech translation (S2ST) [15, 20, 21]. These developments in speech discretization appear to drastically reduce the speech-text modality gap, suggesting that the modality-specific modules employed by soft parameter sharing methods may not be necessary.

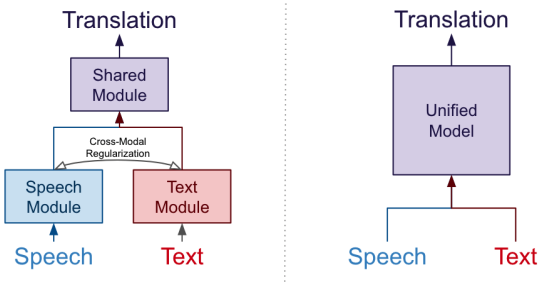

Fig. 1. Illustrative examples of soft (left) vs. hard (right) parameter sharing approaches to ST/MT multi-tasking.

In this work we investigate *hard parameter sharing* [4], where a single unified architecture handles both ST and MT without any modality-specific modules or cross-modal regularization – we posit that this **hard multi-tasking** approach can:

1. Be generally applicable to any sequence-to-sequence model
2. Be used to incorporate external MT data
3. Improve transfer learning from pre-trained textual models

Inspired by the recent AudioPalm work [15] which demonstrates hard ST/MT multi-tasking via a decoder-only model,[2] we investigate a similar concept for attentional encoder-decoder (AED), Connectionist Temporal Classification (CTC), transducer (RNN-T), and joint CTC/attention (CTC/Attn) models. Our method pre-processes speech and text inputs to produce two discrete sequences of comparable length; **doing so allows ST/MT multi-tasking to be realized through a single token-to-token sequence model**. Our models thus ingest speech and text inputs simply using a joint vocabulary. Intuitively, the speech modality is treated as another "language" represented by a distinct writing system. Specifically, speech is *discretized* via k-means clustering over SSL representations (e.g. WavLM [22]) and *down-sampled* via repetition removal and subword tokenization while text is *up-sampled* via token repetition.

In our experiments, we first show that our hard parameter sharing approach improves AED, CTC, RNN-T, and CTC/Attn models when trained from scratch by an average of +0.5 BLEU on the

[1]Recipes and models are available in ESPnet (egs2/must_c_v2/st2).

[2]Please refer to §5 for a full accounting of the novelties in this work.

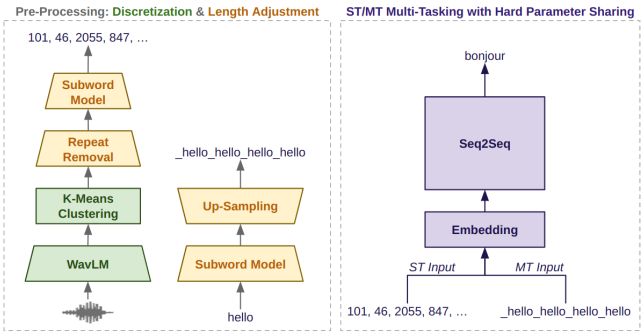

Fig. 2. Speech/text pre-processing (left) produces two discrete sequences of similar length which are ingested by Seq2Seq models (right) with hard parameter sharing between ST/MT tasks.

MuST-C ST corpus [23] without any external MT data (§4.1). Next, we show that leveraging external WMT [24] MT data via our multi-tasking framework yields an additional +0.8 BLEU (§4.2). Finally, we show that our multi-tasking approach also improves the efficacy of transfer learning by +1.8 BLEU from pre-trained textual models (e.g. mBART [25, 26]) (§4.3).

2. PROPOSED FRAMEWORK

In this section, we first describe our method of converting continuous speech into sequences of discrete units (§2.1) and then explain how discrete speech inputs enable **hard parameter sharing** (§2.2). §2.3 describes the set of sequence-to-sequence (seq2seq) frameworks which we investigate.

Figure 2 also summarizes our approach. The color-coding in the figure matches bolded keywords in the following section.

2.1. Speech Discretization

The objective of speech discretization is to convert a continuous speech signal, $X^{\text{CONT}} = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, ..., T\}$, into a discrete sequence, $X^{\text{DISC}} = \{x_l \in \mathcal{V}^{\text{SPE}} | l = 1, ..., L\}$ where $\mathcal{V}^{\text{SPE}}$ is a vocabulary consisting of some discrete units representing chunks of speech. There are many discretization techniques which can accomplish this [18, 27–29]; we opt for the approach described by [14] which uses k-means clustering and SSL representations in a manner similar to HuBERT [30].

We first select an appropriate SSL model and intermediate layer to work with. In this work, we use **WavLM** which is pre-trained on 94k hours of English speech with masked prediction and denoising objectives [22]. We select the 21st layer from WavLM as it has the highest canonical correlation analysis (CCA) similarity to word labels [31], suggesting that these representations contain useful semantic information. Next, we extract these representations for a portion of our training data and use them to train a k-means model with 2000 centroids. We then use this k-means model to convert the entire training set into sequences of **k-means cluster** assignments; note that at this point we have a sequence of discrete units but have only down-sampled the sequence length to 50 kHz via WavLM.

As noted by prior works [14, 20], these sequences of k-means cluster assignments can be collapsed by **removing repeats** of the same consecutive unit. Finally, **subword modeling** can be applied to further reduce the sequence length [14]; we use the unigram algorithm from SentencePiece [32] to construct a vocabulary of 4000 tokens. To reduce over-fitting on particular segmentation patterns, we also apply BPE-dropout [33] during training. Ultimately, we obtain discrete speech sequences with an average length of 122 tokens from original audio with an average duration of 6.4 seconds.

2.2. ST/MT Multi-Tasking with Hard Parameter Sharing

2.2.1. *ST from Discrete Speech Inputs*

With discrete speech inputs, the ST task seeks to map a sequence of tokens X^{DISC} into another sequence of tokens $Y^{\text{TGT}} = \{y_m^{\text{TGT}} \in \mathcal{V}^{\text{TGT}} | l = 1, ..., M\}$ where $\mathcal{V}^{\text{TGT}}$ is the subword vocabulary for the target language – this is analogous to the MT task. We therefore replace the convolutional feature extractor typically used by systems which process continuous speech inputs [34] with a learned **embedding** layer [35]. Following [14], we apply time-masking to the sequences produced by the embedding layer; this is an additional form of data augmentation similar to SpecAugment [36] which is commonly applied to continuous speech inputs.

Our ST data is defined as a set of speech, source text, and target text triplets $\{(X^{\text{DISC}}, Y^{\text{SRC}}, Y^{\text{TGT}})\}$. Similar to target text, source text is a sequence of tokens $Y^{\text{SRC}} = \{y_n^{\text{SRC}} \in \mathcal{V}^{\text{SRC}} | l = 1, ..., N\}$ where $\mathcal{V}^{\text{SRC}}$ is the subword vocabulary for the source language.

2.2.2. *Incorporating the MT Task*

Since the discrete speech sequences X^{DISC} are still longer than their corresponding source text Y^{SRC}, we repeat the source text tokens by a factor of 4 to further reduce the gap between speech and text inputs (see §4.4 for **up-sampling** factor ablations). For instance, the sequence "_a_b" becomes "_a_a_a_a_b_b_b_b". This length adjustment approach has been shown to be effective for injecting text into speech models [37, 38]. We denote this up-sampled source text as X^{TEXT} and define our MT data as triplets of up-sampled source text, source text, and target text $\{(X^{\text{TEXT}}, Y^{\text{SRC}}, Y^{\text{TGT}})\}$.

To incorporate textual inputs into our discrete ST models, we simply extend the input vocabulary $\mathcal{V}^{\text{CROSS}} = \mathcal{V}^{\text{SPE}} \cup \mathcal{V}^{\text{SRC}}$ to include textual subword tokens $\mathcal{V}^{\text{SRC}}$ from the source language in addition to the speech subword tokens $\mathcal{V}^{\text{SPE}}$. This modification correspondingly expands the **embedding** layer, but does not impact any other component in the architecture. Note that this joint speech-text vocabulary allows our models to indiscriminately ingest speech or text into any **seq2seq** model, sharing all non-embedding parameters between ST and MT. The only modality-specific parameters are within the embedding layer, as speech and text tokens are still disjoint.

Now MT multi-tasking can be achieved by simply combining ST and MT training data: the training set consists of triplets $\{(X^{\text{CROSS}}, Y^{\text{SRC}}, Y^{\text{TGT}})\}$ where $X^{\text{CROSS}} = \{x_l \in \mathcal{V}^{\text{CROSS}} | l = 1, ..., L\}$. Note that the source text Y^{SRC} and the target text Y^{TGT} are identical for ST and MT examples. The same losses (described in the following section) are applied with equal weighting between the two tasks. *All* parameters are updated in each iteration. Models do not have any explicit sense of whether a particular example is an MT or ST task – all are processed in the same manner.

2.3. Seq2Seq Models

In this work, we examine AED [39, 40], CTC [41], RNN-T [42], and CTC/Attn [43, 44] models. We use a hierarchical encoding scheme, as in [44], for all of our models. This method applies an ASR CTC objective at an intermediate encoder layer, denoted as $\mathcal{L}_{\text{SRC\_CTC}}$, and a second ST CTC objective at the final encoder layer,

denoted as $\mathcal{L}_{\text{TGT\_CTC}}$. The ASR CTC objective allows our models to utilize source language transcriptions to improve encoder representations [45, 46]. The ST CTC objective acts as a form of regularization which encourages encoder representations to be monotonic with respect to the target sequence; this has been shown to improve the translation quality of auto-regressive systems [44, 47]. Our AED and CTC/Attn models use an additional cross-entropy loss, denoted as $\mathcal{L}_{\text{CE}}$, while our RNN-T models use an additional RNN-T loss, denoted as $\mathcal{L}_{\text{RNNT}}$. AED and RNN-T models are jointly trained with CTC losses but CTC likelihoods are not applied during decoding. All told, our models are optimized using an interpolated loss defined as $\mathcal{L} = \lambda_1\mathcal{L}_{\text{SRC\_CTC}} + \lambda_2\mathcal{L}_{\text{TGT\_CTC}} + \lambda_3\mathcal{L}_{\text{CE/RNNT}}$. We use $\lambda_1 = \lambda_2 = 0.3$ and $\lambda_3 = 1$ for our experiments. For CTC models, the last term is omitted and we use $\lambda_1 = \lambda_2 = 1$.

3. EXPERIMENTAL SETUP

We compare the performance of our **hard multi-task models** vs. **single-task baselines** with identical architectures. The single-task baselines are also discrete ST systems which allow us to understand the effects of ST/MT multi-tasking, holding all else equal. Note that the purpose of this work is not to prove the efficacy of systems with discrete speech inputs compared to those with continuous spectral inputs; this aspect has been addressed elsewhere [14]. We use the ESPnet-ST-v2 toolkit [48] for our experiments.

Data: We use the En-De, En-Es, and En-Fr portions of MuST-C [23] which consist of 408/504/492 audio hours and 234K/270K/292K sentences. Experiments using external MT data were conducted on the En-De language pair using the WMT'16 corpus [24] which consists of 4.6M sentences. Speed perturbation is applied to up-sample ST data by a factor of 3.

Models: We use separate vocabularies of 4000 subword units built from MuST-C data for discrete speech, source text, and target text. Unified multi-tasking models use a combined speech-text vocabulary consisting of 8000 units, obtained by combining discrete speech and source text vocabularies. Models with mBART initializations adopt the pre-trained model's 250K target vocabulary.

All models use input embedding with 1024 dim. We use 18 layer E-Branchformer [49] encoders with ASR CTC applied on the 12th layer. Our base size models (denoted by `A-D` the following section) use 256 dim size, 1024 feed-forward dim, and 4 heads. Our larger models (denoted by `E-F` the following section) use 512 dim size, 2048 feed-forward dim, and 8 heads. We use 6 layer Transformer decoders for AED and CTC/Attn models with 2048 feed-forward dim and either 4/8 heads for base/large models. Finally, for models with mBART we initialize only the decoder while freezing feed-forward and self-attention parameters and use 2x convolutional down-sampling after the encoder, following [50, 51].

To ensure fair comparison, all models are trained for the same number of iterations regardless the training data size. All model converge within 350K iterations and we average the 10 best checkpoints. AED, RNN-T, and CTC/Attn models use beam search with beam size 10. CTC models use greedy decoding.

Evaluation: We measure detokenized case-sensitive BLEU [52].

4. RESULTS AND DISCUSSION

The objective of this work is to study several related, yet still distinct, dynamics within unified ST/MT multi-tasking. We examine the effects of 1) **hard parameter sharing** for a set of sequence models (§4.1), 2) leveraging **external MT data** (§4.2), and 3) **transfer learning** from pre-trained textual models (§4.3). We also provide ablations over sequence lengths (§4.4).

Table 1. Comparison (BLEU scores) of single-task vs. hard ST/MT multi-task approaches for CTC, RNN-T, AED, CTC/Attn models.

#	MODEL	SIZE	MuST-C En-De	MuST-C En-Es	MuST-C En-Fr	MuST-C avg
	CTC					
A1	Single-Task	50M	23.2	27.9	32.2	27.8
A2	**Hard Multi-Task**	50M	**23.4**	**28.4**	**33.6**	**28.5**
	RNN-T					
B1	Single-Task	60M	26.4	30.4	33.1	30.0
B2	**Hard Multi-Task**	60M	**26.7**	**31.0**	**33.8**	**30.5**
	AED					
C1	Single-Task	60M	27.4	32.8	37.4	32.5
C2	**Hard Multi-Task**	60M	**27.7**	**33.1**	**38.0**	**33.0**
	CTC/ATTN					
D1	Single-Task	60M	28.6	33.0	38.7	33.4
D2	**Hard Multi-Task**	60M	**29.2**	**33.2**	**39.2**	**33.9**

4.1. Hard Parameter Sharing

As an alternative to soft parameter sharing methods which manage the "distance" between speech and text representations [8–12], we employ a hard parameter sharing approach which uses a single set of parameters to capture both tasks. To examine the effect of hard parameter sharing, we train models *from scratch* and *without external MT data* in this section. Per §2.2.2, we create an MT example $(X^{\text{TEXT}}, Y^{\text{SRC}}, Y^{\text{TGT}})$ from each ST example $(X^{\text{DISC}}, Y^{\text{SRC}}, Y^{\text{TGT}})$ and combine all ST/MT examples to form a training set.

The results in Table 1 compare our hard multi-tasking method compared to the single-task baseline for CTC, RNN-T, AED, and CTC/Attn models. We observe consistent improvements in the range of +0.2 to +1.4 BLEU points across three different language pairs and for all model types. We attribute these improvements to primarily to the regularization effect of hard parameter sharing [4], as we do not explicitly tell the model how to relate corresponding text and speech inputs. Note that we observed the same trend when using the hierarchical encoding scheme (described in §2.3) as we did without; however, since this scheme produced better translation quality for all models, we chose to only present those results.

4.2. Leveraging External MT Data

The ability to add external MT data into the training mixture is a major benefit of ST/MT multi-tasking. Our approach is simple: we *simply concatenate data sources* and train on the combined set. The first two horizontal partitions of Table 2 present results on English-to-German ST with 4.6M sentences of external MT data from WMT'16. Comparing `E1` to `E2+`, we see the full effect of multi-tasking with external MT data: +1.1 BLEU. A portion of this gain must be attributed to that of hard parameter sharing. To understand the impact of the external MT data on its own, we compare multi-tasking without external data, `E2`, to multi-tasking with external data, `E2+`: +0.8 BLEU. Note that we can pre-train our entire model on this same external MT data, but this constitutes a form of transfer learning which we will discuss in the subsequent section.

Table 2. Performance of hard ST/MT multi-task CTC/Attn models with external MT data or mBART initialization. Single-task CTC/Attn baselines and soft multi-task models from prior works are shown for comparison. †Uses WMT16 MT data (4.6M sentences). ‡Uses external MT and unpaired text data via mBART initialization.

#	MODEL	SIZE	EXT DATA MT	EXT DATA Text	En-De
-	ConST'22 [8]	150M	-	-	25.7
-	M^{3}ST'23 [11]	-	-	-	26.4
E1	Single-Task CTC/Attn	190M	-	-	29.0
E2	**Hard Multi-Task CTC/Attn**	190M	-	-	**29.3**
-	ConST'22 [8]	150M	✓†	-	28.3
-	M^{3}ST'23 [11]	-	✓†	-	29.3
E2+	**Hard Multi-Task CTC/Attn**	190M	✓†	-	**30.1**
F1	Single-Task CTC/Attn	740M	✓‡	✓‡	29.3
F2	**Hard Multi-Task CTC/Attn**	740M	✓‡	✓‡	**31.1**

Table 3. Ablation study on the importance of up-sampling MT input text. MT = Use of MT multi-task. Up = Up-sampling factor of MT input text. Ratio = Average length ratio of discrete speech to text.

MT	Up	Ratio	BLEU
-	-	-	28.6
✓	-	6.0	28.5
✓	2x	3.0	28.8
✓	4x	1.5	**29.2**
✓	6x	1.0	28.7

We take two representative works for comparison in this section. The ConST model [8] is a soft multi-tasking approach which utilizes a contrastive loss to encourage matched speech and text inputs to be closer, relative to unmatched speech and text inputs. ConST also uses multiple strategies to create harder examples for the contrastive loss. The M^{3}ST model [11] is another soft multi-tasking approach which utilizes a multi-stage training strategy. The first stage is a purely textual pre-training stage which incorporates external MT data while the next two stages are ST fine-tuning stages which perform data mix-up and contrastive learning. ConST and M^{3}ST appear to gain more from the same external MT data, although their baselines are indeed much weaker. Nonetheless, we suspect that modality-specific modules can limit interference from extremely unbalanced MT to ST data ratios, but we leave this for future work.

4.3. Cross-Modal Transfer Learning

Ultimately, we'd like to build ST models which *efficiently* leverage not only paired textual data, but also copious amounts of *unpaired textual data*. In this section we examine models initialized from mBART [25, 26], an encoder-decoder pre-trained with text denoising objectives and then fine-tuned on large-scale MT data.[3] The recent trend is to take the mBART decoder parameters to partially initialize ST encoder-decoder models [50, 51, 53, 54]. This is a form of *heterogeneous* transfer learning [55] – there is a cost associated with the distributional shift between the textual pre-training domain and the speech-based fine-tuning domain. Cross-modal pre-training [6, 7] has been shown to reduce this cost. We posit that our cross-modal fine-tuning method has a similar effect.

[3]https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

The results in the final horizontal partition of Table 2 presents models with mBART decoder initialization (see §3). Comparing the single-task model F1 with the multi-task model F2, we see that the latter is +1.8 BLEU better. The single-task model only improves by +0.3 BLEU from mBART initialization (E1 vs. F1); prior works have also noted similarly muted gains [51, 56], indicating deficiencies in transfer learning across modalities. Our method exhibits a more efficient transfer (E2 vs. F2), yielding +1.8 BLEU. Note that mBART has been fine-tuned on large scale MT data, so we do not find it necessary to include WMT data in our training mixture.

4.4. Ablations on Sequence Lengths

Table 3 shows an ablation study on the importance of up-sampling the lengths of MT inputs to match the lengths of discrete speech inputs (per §2.2.2). We found that 4x up-sampling was best. Note that without any up-sampling, ST/MT multi-tasking was actually slightly detrimental. 6x up-sampling was not the best even though speech and text have equal lengths on average, suggesting that the alignment of each text token to corresponding speech tokens is not uniform.

5. RELATION TO PRIOR WORK

Now that we have presented our approach and results, we'll highlight the technical and empirical **novelty** of our work.

First, our work is closely related to AudioPalm [15]; both of our methods achieve hard ST/MT multi-tasking by discretizing speech. On the surface this makes our methods look quite similar, but AudioPalm focuses on initializing speech models from Palm. In fact, their results show that their models are deficient when trained from scratch (see Table 6 in their paper). We take the exact opposite approach: we first confirm that our method improves training from scratch before adding external MT data and initialization from textual pre-trained models. This is a major technical difference in itself, but it also allows us to investigate several empirical novelties. Namely, we are able to show the individual effects of hard parameter sharing, external MT data, and transfer learning. These three effects are conflated within the experimental setup of the AudioPalm paper which is more focused on demonstrating performance at scale. All told, we view these works as complementary – this work focuses on a set of sequence-to-sequence models commonly used in ST and other speech processing tasks (CTC, AED, CTC/Attn, RNN-T) which are distinct from AudioPalm's decoder-only model.

Second, our work follows a long line of prior works which investigate ST/MT multi-tasking [5–13]. The common theme amongst these approaches is soft parameter sharing, which is a major difference compared to our approach. Further, we examine a larger set of Seq2Seq models to demonstrate general applicability.

6. CONCLUSION

We present a method for ST/MT multi-tasking with hard parameter sharing, which is not trivially achieved due to the speech-text modality gap. Our approach resolves this by pre-processing speech into discrete sequences of tokens. This allows us to build Seq2Seq models capable of ingesting speech and text via an input vocabulary consisting of discrete speech and text tokens. Given the consistent improvements in ST, we will apply this approach to spoken language understanding and speech summarization in the future.

Brian and Shinji are supported by the HLTCOE at JHU. This work used NCSA Delta (project CIS210014) from ACCESS through NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296.

7. REFERENCES

[1] J. Pino *et al.*, "Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade," *Proc. IWSLT*, 2019.

[2] H. Inaguma, T. Kawahara, and S. Watanabe, "Source and target bidirectional knowledge distillation for end-to-end speech translation," *Proc. NAACL*, 2021.

[3] Y. Jia *et al.*, "Leveraging unsupervised and weakly-supervised data to improve direct speech-to-speech translation," *Interspeech*, 2022.

[4] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[5] J. Ao *et al.*, "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing," *Proc. ACL*, 2021.

[6] A. Bapna *et al.*, "Mslam: Massively multilingual joint pre-training for speech and text," *arXiv preprint arXiv:2202.01374*, 2022.

[7] Y. Tang *et al.*, "Unified speech-text pre-training for speech translation and recognition," in *Proc. ACL*, 2022.

[8] R. Ye, M. Wang, and L. Li, "Cross-modal contrastive learning for speech translation," in *Proc. NAACL*, 2022.

[9] Q. Fang *et al.*, "STEMM: Self-learning with speech-text manifold mixup for speech translation," in *Proc. ACL*, 2022.

[10] Z. Chen *et al.*, "Maestro: Matched speech text representations through modality matching," *Proc. Interspeech*, 2022.

[11] X. Cheng *et al.*, "M 3 st: Mix at three levels for speech translation," in *Proc. ICASSP*, 2023.

[12] S. Ouyang, R. Ye, and L. Li, "WACO: Word-aligned contrastive learning for speech translation," in *Proc. ACL*, 2023.

[13] J. Wu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," *arXiv:2307.03917*, 2023.

[14] X. Chang *et al.*, "Exploration of Efficient End-to-End ASR using Discretized Input from Self-Supervised Learning," in *Proc. Interspeech*, 2023.

[15] P. K. Rubenstein *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[16] T. Hayashi and S. Watanabe, "Discretalk: Text-to-speech as a machine translation problem," *arXiv preprint arXiv:2005.05525*, 2020.

[17] K. Lakhotia *et al.*, "On generative spoken language modeling from raw audio," *TACL*, 2021.

[18] C. Wang *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[19] M. Hassid *et al.*, "Textually pretrained speech language models," *arXiv preprint arXiv:2305.13009*, 2023.

[20] A. Lee *et al.*, "Direct speech-to-speech translation with discrete units," in *Proc. ACL*, 2022.

[21] X. Li, Y. Jia, and C.-C. Chiu, "Textless direct speech-to-speech translation with discrete speech representation," in *Proc. ICASSP*, 2023.

[22] S. Chen *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *J-STSP*, 2022.

[23] M. A. Di Gangi *et al.*, "Must-c: A multilingual speech translation corpus," in *Proc. NAACL*, 2019.

[24] O. Bojar *et al.*, "Findings of the 2016 conference on machine translation," in *Proc. WMT*, 2016.

[25] Y. Liu *et al.*, "Multilingual denoising pre-training for neural machine translation," *TACL*, 2020.

[26] Y. Tang *et al.*, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv preprint arXiv:2008.00401*, 2020.

[27] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Proc. Neurips*, 2017.

[28] N. Zeghidour *et al.*, "Soundstream: An end-to-end neural audio codec," *TASLP*, 2021.

[29] A. Défossez *et al.*, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[30] W.-N. Hsu *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *TASLP*, 2021.

[31] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *Proc. ICASSP*, 2023.

[32] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, 2018.

[33] I. Provilkov, D. Emelianenko, and E. Voita, "Bpe-dropout: Simple and effective subword regularization," in *Proc. ACL*, 2020.

[34] R. Prabhavalkar *et al.*, "End-to-end speech recognition: A survey," *arXiv preprint arXiv:2303.03329*, 2023.

[35] F. Stahlberg, "Neural machine translation: A review," *Journal of Artificial Intelligence Research*, 2020.

[36] D. S. Park *et al.*, "Specaugment: Simple data augmentation for automatic speech recognition," *Proc. Interspeech*, 2019.

[37] A. Renduchintala *et al.*, "Multi-modal data augmentation for end-to-end asr," *Proc. Interspeech*, 2018.

[38] S. Thomas *et al.*, "Integrating text inputs for training and adapting rnn transducer asr models," in *Proc. ICASSP*, 2022.

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[40] W. Chan *et al.*, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[41] A. Graves *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[42] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *Proc. ICML*, 2012.

[43] S. Watanabe *et al.*, "Hybrid ctc/attention architecture for end-to-end speech recognition," *JSTSP*, 2017.

[44] B. Yan *et al.*, "Ctc alignments improve autoregressive translation," in *Proc. EACL*, 2023.

[45] H. Inaguma *et al.*, "Espnet-st: All-in-one speech translation toolkit," in *Proc. ACL*, 2020.

[46] M. Gaido *et al.*, "Ctc-based compression for direct speech translation," in *Proc. EACL*, 2021.

[47] B. Zhang, B. Haddow, and R. Sennrich, "Revisiting end-to-end speech-to-text translation from scratch," in *Proc. ICML*, 2022.

[48] B. Yan *et al.*, "ESPnet-ST-v2: Multipurpose spoken language translation toolkit," in *Proc. ACL*, 2023.

[49] K. Kim *et al.*, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. SLT*, 2023.

[50] X. Li *et al.*, "Multilingual speech translation from efficient finetuning of pretrained models," in *Proc. ACL*, 2021.

[51] B. Yan *et al.*, "Cmu's iwslt 2023 simultaneous speech translation system," in *Proc. IWSLT*, 2023.

[52] M. Post, "A call for clarity in reporting bleu scores," in *Proc. WMT*, 2018.

[53] N.-Q. Pham *et al.*, "Effective combination of pretrained models-kit@ iwslt2022," *IWSLT 2022*, 2022.

[54] Z. Zhang *et al.*, "The yitrans end-to-end speech translation system for iwslt 2022 offline shared task," *Proc. IWSLT*, 2022.

[55] O. Day and T. M. Khoshgoftaar, "A survey on heterogeneous transfer learning," *Journal of Big Data*, 2017.

[56] H. Inaguma *et al.*, "UnitY: Two-pass direct speech-to-speech translation with discrete units," in *Proc. ACL*, 2023.