WACO: Word-Aligned Contrastive Learning for Speech Translation

Siqi Ouyang[1], Rong Ye[2], Lei Li[1]
[1]University of California, Santa Barbara, CA, USA
siqiouyang@ucsb.edu, leili@cs.ucsb.edu
[2]ByteDance AI Lab, Shanghai, China
yerong@bytedance.com

Abstract

End-to-end Speech Translation (E2E ST) aims to directly translate source speech into target text. Existing ST methods perform poorly when only extremely small speech-text data are available for training. We observe that an ST model's performance closely correlates with its embedding similarity between speech and source transcript. In this paper, we propose **W**ord-**A**ligned **CO**ntrastive learning (**WACO**), a simple and effective method for extremely low-resource speech-to-text translation. Our key idea is bridging word-level representations for both speech and text modalities via contrastive learning. We evaluate WACO and other methods on the MuST-C dataset, a widely used ST benchmark, and on a low-resource direction Maltese-English from IWSLT 2023. Our experiments demonstrate that WACO outperforms the best baseline by 9+ BLEU points with only 1-hour parallel ST data. Code is available at https://github.com/owaski/WACO.
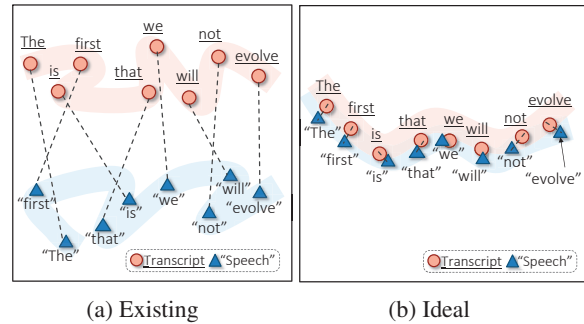
(a) Existing (b) Ideal

Figure 1: Better representations of speech and transcript text (projected to 2D) will lead to improved speech translation. Ideal representations (b) should not only be similar at the sequence level, but also closely align corresponding words between speech and transcript.

1 Introduction

End-to-end speech translation (E2E ST) directly translates speech in a source language to text in a target language, without intermediate pipelines. E2E ST has witnessed significant progress in translation quality (Duong et al., 2016; Weiss et al., 2017; Bérard et al., 2018; Jia et al., 2019; Gaido et al., 2020; Dong et al., 2021a; Ye et al., 2022a) . However, existing E2E ST methods still perform poorly when only a limited amount of parallel ST data are available. How can we build a highly performant ST model with extremely low resource, e.g. only 1 hour of parallel data (approximately a few hundred utterances)?

We analyze the encoder representations from a directly learned ST model[1] and find that its average embeddings from speech and transcript are similar at the sequence level but still not aligned

well on the word level (Figure 1a). An ideal ST model should encode a speech utterance closely aligned with the representations of corresponding words in its transcript text (Figure 1b). Prior methods attempt to use additional parallel data from machine translation (MT) and automatic speech recognition (ASR) to align speech and text representations. However, most do not explicitly reduce the word level representation gap among speech and text. We hypothesize that such misalignment at the word level between corresponding speech and transcript text is a critical cause of the inferior ST performance.

We further observe such a misalignment phenomenon is severe when ST data is extremely low. We did a pilot study by training direct ST models using different sizes of ST data (1/5/10/388 hours) . As shown in Figure 2, we find that the translation performance highly correlates with the word-level embedding similarity between speech and transcript text. With fewer parallel ST data, the cross-modal similarity drops simultaneously with the BLEU score and almost reaches 0 given 1-hour ST training data. This observation suggests that the model can map both modalities into a partially aligned semantic space given sufficiently large ST

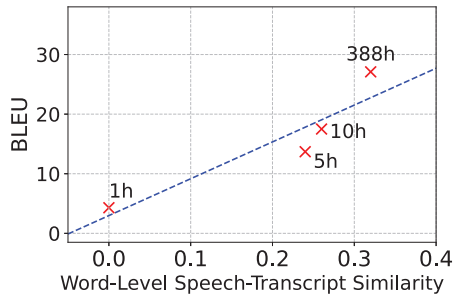[1]We use XSTNet (Ye et al., 2021) as an example.

Figure 2: BLEU score of ST models trained on varying amounts of ST data (1/5/10/388 hours) and their cosine similarity scores between speech and transcript word embeddings. The ST performance closely correlates with speech-text representation similarity and both drop significantly when ST data is extremely limited.

data but fails when ST data is extremely small.

In this paper, we propose WACO, a word-level contrastive learning method for extremely low-resource speech-to-text translation. Its key idea is to promote representation similarity among corresponding words in speech and text and to push non-corresponding representations further apart in embedding spaces. Furthermore, it can use additional large ASR data to improve word-level representation learning. We conduct experiments on the MuST-C benchmark with varying data sizes and a real-world low-resource direction Maltese-English (Mt-En) from IWSLT 2023 low-resource track. We also analyze the word-level representation similarity from the learned models. Our contributions are:

- We propose a new method for speech translation, WACO, which explicitly aligns speech-transcript representations of corresponding words.
- We verify the effectiveness of WACO on MuST-C and IWSLT Mt-En. WACO outperforms previous strong methods by 2.0-9.8 BLEU points with only 1 and 10 hours of parallel ST data.
- We further demonstrate that that WACO indeed learns a better-aligned representation of speech and text at the word level which correlates well with its ST performance.

2 Related Work

End-to-end ST Due to error propagation and high latency in cascaded ST systems, Bérard et al. (2016); Duong et al. (2016) first proposed to translate source speech into target text directly without generating the intermediate transcript. The major difficulty in training end-to-end ST systems is the lack of direct ST data. Though many ST

datasets (Wang et al., 2021; Cattoni et al., 2021) were proposed in recent years, the amount of ST data is still much less than that of MT and ASR. To overcome the data scarcity problem, methods including data augmentation (Park et al., 2019; McCarthy et al., 2020; Lam et al., 2022; Mi et al., 2022), self-training (Pino et al., 2020), multi-tasking (Le et al., 2020; Tang et al., 2021b,a; Ye et al., 2021; Zhang et al., 2022a) and pre-training (Berard et al., 2018; Bansal et al., 2019; Wu et al., 2020; Wang et al., 2020; Alinejad and Sarkar, 2020; Dong et al., 2021a; Zheng et al., 2021; Bapna et al., 2021; Ao et al., 2022; Tang et al., 2022) have been proposed. WACO is a novel approach that can be applied in existing multi-tasking and pre-training frameworks to improve ST performance.

Cross-modal representation learning Researchers realized recently that the misalignment between speech and text representation hinders the knowledge transfer from external data (Liu et al., 2020; Dong et al., 2021b; Xu et al., 2021; Han et al., 2021; Ye et al., 2022b; Du et al., 2022; Wang et al., 2022). Liu et al. (2020) shrank the speech representation to match the length of text representation and also closed the representational gap by minimizing their L2 distance. Xu et al. (2021) mapped speech representation to text representation through both the Connectionist Temporal Classification (CTC) (Graves et al., 2006) distribution and a mapping layer. Dong et al. (2021b) proposed a cross attention layer to force the speech-text alignment. Han et al. (2021) developed a novel architecture enabling fixed-length shared semantic space for both modalities. Saxon et al. (2021) proposed a hierarchical speech understanding system leveraging both ASR and text understanding data. Ye et al. (2022b) employed sentence-level contrastive loss to reduce the modality gap and ach'ieved state-of-the-art results on MuST-C. Our method, however, works on word-level instead of sentence-level and empirically provides both better performance and higher data efficiency. Fang et al. (2022) also proposes to align the word-level representations between speech and text, but their method heavily relies on target translation while our method only requires ASR data for modality gap reduction.

3 Proposed Method: WACO

In this section, we describe our approach to develop effective speech translation models with extremely low-resource parallel data.

ST & MT Loss   ASR Loss

<de> Hallo Welt!   *<en> Hello world!*

Transformer Encoder-Decoder

Word-Aligned Contrastive Loss

Speech Encoder   Text Embedding

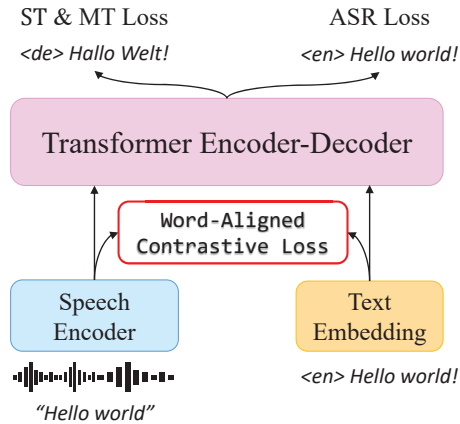*<en> Hello world!*

"Hello world"

Figure 3: Model architecture of WACO. During training, it accepts either speech or text input and outputs text sequence. We introduce word-aligned contrastive loss to align representations of speech and transcript text. At inference time, it only needs speech input.

3.1 Problem Formulation

A typical ST corpus $\mathcal{D}^{\text{ST}}$ contains speech s and its transcript x in a source language and translation text y in a target language. Equivalently, $\mathcal{D}^{\text{ST}} = \{(s, x, y)\}$ and ASR corpus can be similarly defined as $\mathcal{D}^{\text{ASR}} = \{(s, x)\}$.

Given $\mathcal{D}^{\text{ST}}$ and $\mathcal{D}^{\text{ASR}}$ as training sets, the E2E ST model needs to translate speech s into translation y accurately without generating transcript x in the intermediate steps. There are two settings:

- **Regular ST**: Training includes large ST triplet data. In this paper, we regard the entire MuST-C training set as the regular setting ($|\mathcal{D}^{\text{ST}}| \approx 400$ hours).
- **Low-Resource ST**: Training has very limited ST data but plenty of ASR data, i.e., $|\mathcal{D}^{\text{ST}}| \ll |\mathcal{D}^{\text{ASR}}|$. In this paper, ST data below 10 hours is regarded as low-resource ST. Many African and native American languages belong to this setting.

In this paper, we focus on low-resource ST. In addition, we also include external MT data for both settings. The size of MT dataset is much larger than ST dataset.

3.2 Model Architecture

Figure 3 illustrates WACO model architecture. WACO consists of 3 modules: a speech encoder, a text embedding layer and a joint Transformer encoder-decoder. During training, it inputs either speech or text sequence and outputs text sequence. At inference time, the input is only speech.

Speech Encoder extracts contextualized acoustic embeddings from the raw waveform. It consists of wav2vec 2.0 (Baevski et al., 2020) and 2 downsampling layers. Wav2vec 2.0 is one of the state-of-the-art self-supervised models pre-trained on unlabeled English speech corpus. It produces contextualized speech embeddings. It consists of 7 convolutional layers as the audio feature extractor and a 12-layer Transformer as the contextual encoder. On top of wav2vec 2.0, we further add 2 convolutional layers with stride of 2 to downsample the embedding sequence. It reduces the length and alleviates the length discrepancy between speech and text embeddings.

Text Embedding is a lookup table that maps text tokens into embeddings.

Transformer Encoder-Decoder accepts outputs from either both the speech encoder or the text embedding layer. The configuration is the same as the 6-layer vanilla Transformer (Vaswani et al., 2017). Specifically, the encoder further extracts contextualized high-level semantic features from both modalities and the decoder generates a token sequence for different tasks. Besides, since we are using general Transformer architecture, both the text embedding layer and the Transformer can be pre-trained on additional MT data.

3.3 Word-Aligned Contrastive Learning

To alleviate the misalignment between speech and transcript representations, we propose word-aligned contrastive learning to bring speech and text embeddings closer in a fine-grained level (Figure 4).

Given a speech-transcript pair (s, x). The transcript is tokenized by a Byte-Pair-Encoding (BPE) tokenizer into a sequence of BPE tokens $x = (x_1, x_2, \cdots, x_n)$. Though BPE is the default option for text tokenization, whole words preserves acoustic boundary better than BPE tokens. Thus, we group n BPE tokens back into m whole words where $w_i = x[l_i^t : r_i^t]$ for $i = 1, 2, \cdots, m$ where superscript t stands for text feature and l_i, r_i refer to the range of BPE tokens that word i is tokenized into. In the example of Figure 4, $x = (\_\text{Pra}, \text{c}, \text{tic}, \text{e}, \_\text{makes}, \_\text{perfect}, .)$, and $w_1 = (\_\text{Pra}, \text{c}, \text{tic}, \text{e})$, $w_2 = (\_\text{makes})$, $w_3 = (\_\text{perfect})$, and $w_4 = (.)$. Then we align whole words $w_1, w_2, \cdots, w_m$ with speech $s = (s_1, s_2, \cdots, s_{|s|})$ by a forced aligner. Here we use Montreal Forced Aligner[2] (MFA).

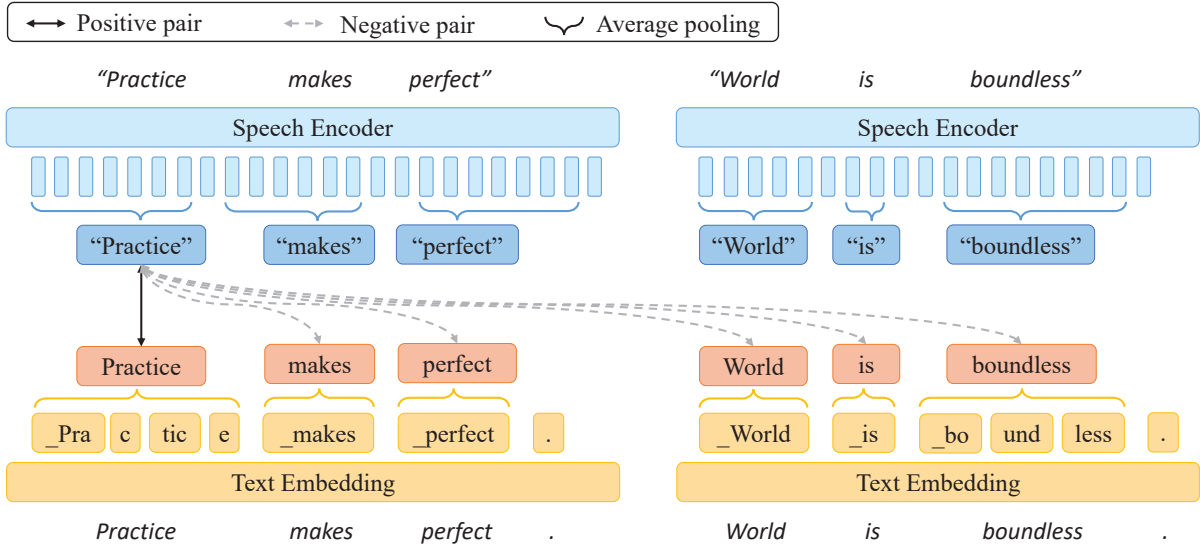[2] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

Figure 4: An illustration of word-aligned contrastive learning for a batch of two data points. Speech and text are passed through speech encoder and text embedding respectively to obtain embeddings. Then we group embeddings by word-level average pooling for both modalities. Average speech and text embeddings for the same word are treated as the positive pair and average embeddings for different words are treated as the negative pairs.

This provides us time interval $1 \leq l_i^s \leq r_i^s \leq |s|$ for word w_i where superscript s stands for speech feature.

Now we have identified m corresponding pairs of speech segments $s[l_i^s : r_i^s]$ and words $x[l_i^t : r_i^t]$ for $i = 1, 2, \cdots, m$. The representations of them are obtained as follows,

$$f_i^s = \text{MeanPool}(\text{S-Enc}(s)[\tilde{l}_i^s : \tilde{r}_i^s]) \quad (1)$$
$$f_i^t = \text{MeanPool}(\text{T-Emb}(x)[l_i^t : r_i^t]) \quad (2)$$

where S-Enc is the speech encoder, T-Emb is the text embedding layer, $\tilde{l}_i^s = \frac{l_i^s}{|s|}|\text{S-Enc}(s)|$ and $\tilde{r}_i^s = \frac{r_i^s}{|s|}|\text{S-Enc}(s)|$ refer to the relative indices given the audio representation length shrinkage after the speech encoder.

We treat f_i^s and f_i^t as a positive pair and treat f_i^s and other words in the same batch as negative pairs and we apply multi-class N-pair contrastive loss (Sohn, 2016) on them:

$$\ell_{\text{CTR}}(\mathcal{B}) =$$
$$- \mathop{\mathbb{E}}_{f_i^s, f_i^t \in \mathcal{B}} \left[\log \frac{\exp(sim(f_i^s, f_i^t)/\tau)}{\sum_{f_j^t \neq i \in \mathcal{B}} \exp(sim(f_i^s, f_j^t)/\tau)} \right] \quad (3)$$

where $\mathcal{B}$ is the current batch, τ is the temperature hyper-parameter, $sim()$ is used to measure the distance between two representations, and we use cosine similarity $sim(a, b) = a^\top b / \|a\|\|b\|$.

3.4 Training and Inference

Cross-Modal Pre-training We first pre-train text embedding and joint Transformer on external MT dataset (e.g., WMT dataset). Then we train the MFA model on ASR data $\mathcal{D}^{\text{ASR}}$ to obtain alignments, and further train our ST model using word-aligned contrastive loss on $\mathcal{D}^{\text{ASR}}$

$$\mathcal{L}^{\text{PT}} = \mathop{\mathbb{E}}_{\mathcal{B} \subseteq \mathcal{D}^{\text{ASR}}} [\ell_{\text{CTR}}(\mathcal{B})], \quad (4)$$

where $\ell_{\text{CTR}}(\mathcal{B})$ is defined in Equation 3. The pre-training stage aims to map speech and text embeddings into a shared semantic space using ASR and MT data.

Multi-task Fine-tuning We fine-tune our model using the multi-task cross-entropy losses, as suggested in Ye et al. (2021), and contrastive loss.

$$\mathcal{L}^{\text{FT}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CTR}} \quad (5)$$

where

$$\mathcal{L}_{\text{CE}} = \mathop{\mathbb{E}}_{(s,x,y) \in \mathcal{D}^{\text{ST}}} [\ell_{\text{ST}} + \ell_{\text{MT}} + \ell_{\text{ASR}}] \quad (6)$$
$$\mathcal{L}_{\text{CTR}} = \mathop{\mathbb{E}}_{\mathcal{B} \subseteq \mathcal{D}^{\text{ST}}} [\ell_{\text{CTR}}(\mathcal{B})]. \quad (7)$$

Cross entropy losses are derived directly from the triplet dataset $\mathcal{D}^{\text{ST}}$,

$$\ell_{\text{ST}}(s, y) = -\log P(y|s) \quad (8)$$
$$\ell_{\text{MT}}(x, y) = -\log P(y|x) \quad (9)$$
$$\ell_{\text{ASR}}(s, x) = -\log P(x|s). \quad (10)$$

λ is the hyper-parameter controlling the weight of contrastive loss.

Inference During inference, the model accepts speech frames as input of speech encoder and decodes translation in the target language through beam search. No source transcript text is needed during inference.

4 Experiments

4.1 Datasets

MuST-C We conduct experiments on the MuST-C dataset[3,4] (Di Gangi et al., 2019), one of the largest ST benchmark datasets containing translations from English to 8 languages collected from TED Talks. Each language direction involves around 400 hours of audio recordings. Limited by computing resources, we examine our method on three language directions: English-German/French/Spanish (En-De/Fr/Es).

MuST-C Low-Resource To examine our method in the extremely low-resource settings, we manually create ASR and ST subsets from the MuST-C En-De training set. Specifically, we build 10-hour, 100-hour and 370-hour ASR subsets and 1-hour and 10-hour ST subsets respectively through random sampling.

IWSLT Low-Resource We also evaluate our method on Maltese to English translation in IWSLT 2023 low-resource track[5]. We use the officially provided ST triplets as ST data and build the ASR dataset by combining the the audio-transcript part from the official ST data and Common-Voice (Ardila et al., 2020)[6]. We remove silences and randomly partition the data. Finally we obtain 1 hour of ST triplets and 10 hours of ASR pairs as training and development set and 0.1 hour of ST triplets as test set.

External ASR We also introduce LibriSpeech (Panayotov et al., 2015) as the external ASR dataset. LibriSpeech is the *de facto* public English ASR benchmark[7] containing 960 hours of audiobook speech. We build a 1330-hour English ASR dataset by combining MuST-C and LibriSpeech.

External MT Additionally, we introduce external WMT En-De/Fr/Es datasets (Bojar et al., 2016)

---

[3]Released under CC BY NC ND 4.0 International
[4]Here we refer to MuST-C v1.0.
[5]https://iwslt.org/2023/low-resource
[6]We do not use the officially provided ASR dataset due to its severe audio-transcript misalignment.
[7]Released under CC BY 4.0

for each language direction to pre-train text embedding and Transformer. We also introduce Flores-200 (Costa-jussà et al., 2022), a massively multilingual machine translation dataset, for Mt-En direction. As shown in previous work (Xu et al., 2021; Ye et al., 2021), MT pre-training greatly improves ST performance.

The statistics of datasets above are listed in Appendix A.1.

4.2 Experimental Setups

Model Configurations In MuST-C experiments, we use wav2vec 2.0 base model[8] in our speech encoder which is solely pre-trained on 960-hour English audio. It consists of a 7-layer convolutional feature extractor and 12 Transformer encoder blocks with 768 hidden units. Two down-sampling convolutional layers have kernel size 5, stride size 2 and hidden size 512 or 1024 depending on the Transformer hidden size. For En-De/Fr/Es directions, the Transformer encoder-decoder has 6 encoder and decoder layers with hidden size 512, 2048 FFN hidden units and 8 attention heads. For Mt-En direction, it has 12 layers each with hidden size 1024, 4096 FFN hidden units and 16 attention heads.

Preprocessing The input speech is the raw 16-bit 16kHz mono-channel waveform. We filter speech that is either too long (>480k frames) or too short (<1k frames) out. This results in 388/471/480 hours of speech being retained as ST training data for En-De/Fr/Es directions. We jointly tokenize the transcripts and translations for each language direction using SentencePiece (Kudo and Richardson, 2018) with a vocabulary size set to 10k. To conduct forced alignment required by WACO (see section 3.3), we first remove punctuations and group whole words by identifying special space token in the vocabulary. Then we use the MFA to train forced aligners on $\mathcal{D}^{\text{ASR}}$ to align English speech and words. **The amount of ASR data used to train forced aligner is the same as that used for training WACO.** Due to vocabulary mismatch between the MFA and our SentencePiece model, a small number of speeches and transcripts (e.g., 18h for En-De) cannot be aligned and we simply ignore them when doing contrastive learning.

Training Transformer and text embedding are pre-trained on the external WMT dataset for En-

---

[8]https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt

3895

Direction	En-De							Mt-En
ST Data	1h				10h			1h
ASR Data	10h	100h	370h	1330h	100h	370h	1330h	10h
Base	4.3	4.3	4.3	4.3	17.5	17.5	17.5	-
Base+CTC	0.2	12.6	14.6	14.7	18.3	20.4	20.0	3.0
ConST	3.0	7.3	11.7	13.7	16.9	18.6	19.6	4.0
WACO	**14.1**	**16.2**	**16.6**	**17.5**	**21.0**	**22.7**	**22.9**	**13.3**

Table 1: Case-sensitive detokenized BLEU scores on MuST-C En-De tst-COMMON set and manually created IWSLT Mt-En test set of models pre-trained on ASR data using different cross-modal methods and fine-tuned on ST data. All models share the same W2V2-Transformer architecture. **Base** ignores ASR data and only fine-tunes on ST data, while other baselines pre-train on ASR data using **CTC**, sentence-level contrastive (**ConST**) and word-aligned contrastive (**WACO**) losses.

De/Fr/Es (MT training details can be found in Appendix A.5). For Mt-En direction, we directly initializes the Transformer and text embedding with the NLLB-600M model (Costa-jussà et al., 2022)[9] pre-trained on Flores-200. For both cross-modal pre-training and multi-task fine-tuning, we set contrastive temperature $\tau = 0.2$ and optimize our model by Adam optimizer (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.98$) with learning rate 1e-4 and 25k warm-up steps. After the warm-up, the learning rate is decayed following the inverse square root schedule. The effective batch size is 16 million frames. We set dropout rate to 0.1. For pre-training, we save the checkpoints with the best contrastive loss on the validation set. For fine-tuning, we save the checkpoints with the best BLEU on the validation set and average the last 10 saved checkpoints. Also, we set label smoothing to 0.1 for the cross-entropy losses, $\lambda = 0$ in low-resource ST and $\lambda = 1$ in ST with full data. All models are trained on Nvidia A6000 GPUs.

Inference and Evaluation During inference, we run beam search with beam size 10 and length penalty 0.6/1.0/0.1/0.3 for En-De/Fr/Es and Mt-En directions respectively. For evaluation, we report case-sensitive detokenized BLEU scores on MuST-C tst-COMMON and IWSLT Mt-En test set using sacreBLEU (Post, 2018)[10].

Baselines In low-resource ST settings, we compare our method with three baselines:
- **Base**: This baseline ignores $\mathcal{D}^{ASR}$ and only optimizes cross entropy loss in Equation 6 on $\mathcal{D}^{ST}$.

- **Base+CTC**: This baseline, on top of **Base**, applies CTC loss on $\mathcal{D}^{ASR}$ to align speech and text representations. In particular, we add a linear layer after the speech encoder to predict the text BPE token at each frame and fix its weight with text embedding. We only include CTC with BPE tokenization here since it performs consistently better than its phoneme counterpart (details in Section 5.2).
- **ConST**: This baseline adds a coarse-grained contrastive loss on $\mathcal{D}^{ASR}$ on top of **Base** to reduce modality gap as in Ye et al. (2022b). Instead of word-level alignment, **ConST** conducts contrastive learning on sentence-level average speech and text embeddings. Hyper-parameters are directly borrowed from Ye et al. (2022b).

In regular ST with full MuST-C data, we compare our method with other existing works.

4.3 Main Results

Low-Resource ST Results are shown in Table 1. The ASR data for cross-modal pre-training varies from 10 hours to 1330 hours, and the ST data for multi-task fine-tuning varies from 1 hour to 10 hours. **WACO** consistently outperforms baseline methods in all data configurations and language directions. In particular, our model achieves a BLEU score of 14.1 for En-De and 13.3 for Mt-En with only 1h ST and 10h ASR data and 21.0 for En-De with only 10h ST and 100h ASR data. With 1330h ASR data, WACO even pushes the BLEU score to 17.5 and 22.9. More surprisingly, we find that **WACO** has a further advantage when using less ASR data. When reducing ASR data from 370 hours to 100 hours, the BLEU score increases (**WACO** vs **Base+CTC**, **ConST**) are enlarged from +2.0,+4.9 to +4.0,+8.9 in 1h ST set-

Models	En-De	En-Fr	En-Es
W-Transf. (Ye et al., 2021)	23.6	34.6	28.4
SpeechT5 (Ao et al., 2022)	25.2	35.3	-
FAT-ST (Zheng et al., 2021)	25.5	-	30.8
JT-S-MT (Tang et al., 2021a)	26.8	37.4	31.0
Chimera (Han et al., 2021)	27.1	35.6	-
XSTNet (Ye et al., 2021)	27.8	38.0	30.8
SATE (Xu et al., 2021)	28.1	-	-
STEMM (Fang et al., 2022)	**28.7**	37.4	31.0
ConST (Ye et al., 2022b)	28.3	**38.3**	**32.0**
WACO	28.1	38.1	**32.0**
STPT (Tang et al., 2022)*	29.2	39.7	33.1
SpeechUT (Zhang et al., 2022b)*	30.1	41.4	33.6

Table 2: Case-sensitive detokenized BLEU scores on MuST-C En-De `tst-COMMON` set of models trained on full MuST-C training set. *Note that SpeechUT and STPT leverage more speech (60k hours) or ASR data (100 hours).

ting. This demonstrates that WACO is more efficient than the baseline methods especially in low-resource setting.

Regular ST Results are shown in Table 2. Here we are using the entire MuST-C training set as in previous works to enable fair comparison, which means $\mathcal{D}^{ST}$ has full MuST-C training data. WACO is competitive with previous models such as STEMM and ConST in all three language directions. Note that SpeechUT and STPT achieve the highest BLEU scores in all directions, but both SpeechUT and STPT leverages additional speech data (60k hours) or ASR data (100 hours) and employ a different model architecture.

5 Analysis

5.1 Why is Word-level Contrastive Loss Better than Sentence-level Contrastive Loss?

Intuitively, only aligning the representations between speech and text at the sentence level cannot assure that model captures the accurate word correspondence between these two modalities.

First, we measure the cosine similarity between speech embedding and transcript text embedding using models (ConST and WACO) pre-trained on 370h ASR dataset and fine-tuned on 1h ST dataset. The embeddings of speech and transcript text are calculated based on Equation 1 and 2 respectively. The result is shown in Table 3. WACO achieves more accurate word-level alignment than ConST (0.65 *v.s.* 0.44), which indicates **WACO can handle word-level details inside a sentence better.** Besides, though not optimized for sentence-level alignment, WACO still achieves close sentence-level similarity with ConST (0.30 *v.s.* 0.33).

	CTC	ConST	WACO
Word-Level Sim	0.08	0.44	**0.65**
Sentence-Level Sim	0.13	**0.34**	0.30

Table 3: Average cosine similarity between word-level and sentence-level representations from speech and transcript. ASR and ST data are fixed to 10h and 1h respectively. WACO achieves much higher word-level similarity than ConST and CTC. Though not optimized for sentence-level objective, WACO still reaches close sentence-level similarity with ConST.

Method	100h ASR	370h ASR
CTC	18.3	20.4
CTC$_{Phoneme}$	14.3	19.0
WACO	**21.0**	**22.7**

Table 4: Case sensitive detokenized BLEU scores on MuST-C En-De tst-COMMON of CTC models with BPE (CTC by default) and with phoneme tokenization (CTC$_{Phoneme}$) and WACO. Fine-tuning ST data is fixed at 10h, while ASR data for pre-training is varied. The same MT pre-trained model is applied.

We also show an example in Figure 5 to further demonstrate the importance of such details. From the similarity matrix, we can see that WACO aligns both modalities quite well for all words but ConST struggles on words "that" and "evolve" as highlighted in blue boxes. This directly results in two translation errors of ConST. First, it fails to recover the clause structure implied by "that". Second, it omits "evolve" entirely in the translation. Though ConST still provides higher sentence similarity than WACO, it fails to understand the subtlety inside the sentence. More examples are in Figure 9.

5.2 Why is WACO better than CTC?

WACO treats words as base units to preserve acoustic boundaries and leverage knowledge of the pretrained MT model, while CTC cannot achieve both merits simultaneously. CTC cannot benefit from word tokenization due to its extremely large vocabulary. To preserve acoustic boundaries, CTC requires phoneme or character tokenization. To leverage pre-trained MT model, CTC requires the same tokenization with MT model, i.e., BPE tokenization, but it has no guarantee on well-behaved acoustic boundaries.
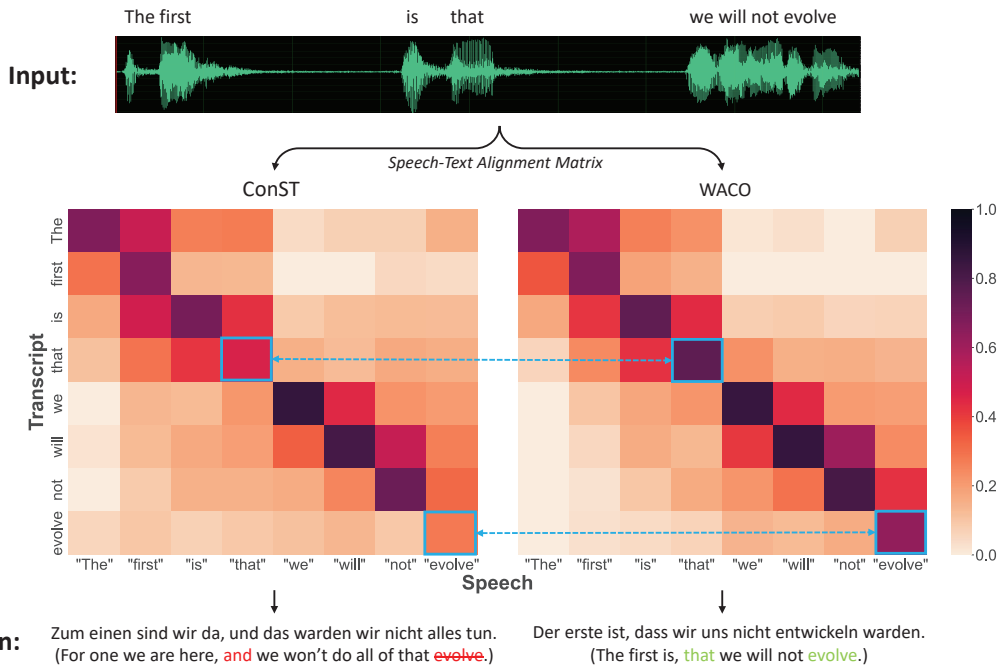
Figure 6 gives a typical example of the token-

Figure 5: An example showing that WACO can capture the word-level details better than ConST. The matrix illustrates pairwise cosine similarity between word-level average embeddings of speech and transcript. WACO aligns two modalities well while ConST fails to align word "that" and "evolve". Though ConST still provides higher sentence-level similarity than WACO (0.60 for ConST and 0.58 for WACO), its translation is not as accurate as our method due to misaligned words.

| MT Data | 4.6M | 4.6M | 0.46M |
W2V2 Init	✓	✗	✓
Cascade ST	15.6	<5	14.1
SeqKD	18.9	18.3	16.9
WACO	14.1	14.3	12.5
w/ SeqKD	**19.5**	**19.2**	**17.5**

Table 5: Case-sensitive detokenized BLEU scores on MuST-C En-De tst-COMMON set of WACO, Cascade ST and Forward Translation. We vary the amount of MT training data from 4.6M to 0.46M and initializes the speech encoder with wav2vec 2.0 or random weights.

to-frame embedding alignment matrix, indicating that WACO learns more accurate alignment compared to CTC with BPE tokenization. Table 3 also substantiates this quantitatively (0.65 and 0.30 *v.s.* 0.08 and 0.13).

For acoustic tokenizations, we evaluate phoneme tokenization. We use the same phoneme vocabulary and grapheme-to-phoneme package as in (Tang et al., 2022). As shown in Table 4, WACO consistently outperforms CTC with phoneme tokenizations, proving its effectiveness in exploiting pretrained MT knowledge.

## 5.3 Comparison with Sequence-Level Knowledge Distillation and Cascade ST

One might argue that for the low-resource scenario we described above, sequence-level knowledge distillation (SeqKD, i.e. forward translation) (Kim and Rush, 2016) and building a cascaded system are the most straightforward way to leverage additional data and improve performance. The former trains an E2E model by expanding the ASR data translated from the existing MT model, while the latter builds the ASR and MT models separately. We compare WACO and these two methods in various settings to demonstrate their differences. The implementation details of SeqKD and Cascade ST are described in Appendix A.6.

We show the result with 10h ASR and 1h ST data in Table 5. Results of more data configurations are shown in Table 10 in the Appendix. We vary the amount of other pre-training resources: speech-only data and MT data. Speech-only data is used to pre-train speech encoder (i.e., wav2vec 2.0) and MT data is used to pre-train text embedding and joint Transformer in WACO and MT models in SeqKD and Cascade ST.

All models drops 1∼2 BLEU points when using 10% MT data. Cascade ST performs better than
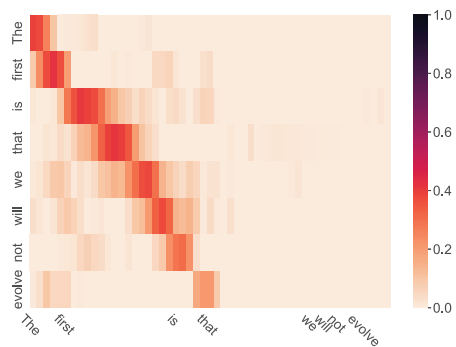
(a) CTC
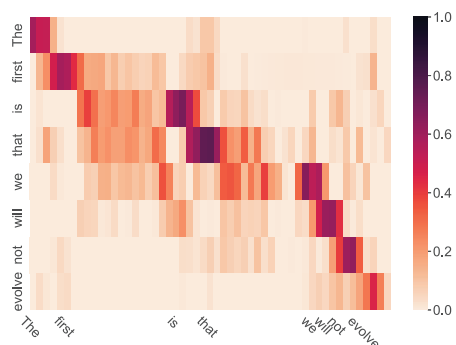
(b) WACO

Figure 6: Token-to-Frame embedding alignment matrix produced by models trained with CTC and WACO respectively. Each row corresponds to a word and each column stands for a frame. Words in X-axis are placed according to their timestamps in speech to show how well the alignments are.

WACO when initialized with wav2vec 2.0. However, WACO outperforms the cascade system when we train speech encoders from scratch. The ASR model trained from scratch in Cascade ST fails to produce meaningful transcript due to severe overfitting. Surprisingly, both WACO and SeqKD are not sensitive to wav2vec 2.0 initialization. WACO and SeqKD are also complementary to each other, and combining them leads to the highest BLEU score.

6   Conclusion

In this work, we propose WACO to align word-level speech and text embeddings. Experiments demonstrate the effectiveness of our method in both low-resource and regular ST settings. Analysis shows that our method can achieve better speech-text alignment which correlates well with its ST performance.

Limitations

There are two main limitations in this work.

First, instead of best ST performance given full data, our cross-modal pre-training only aims to demonstrate the effectiveness of our method in the low-resource ST setting. We realize that unified pre-training for both speech and text gradually becomes a dominant paradigm for ST and our future work is to fuse WACO into a joint pre-training framework.

Second, we note that Tang et al. (2022) explores the possibility of pre-training MT models with phoneme tokenizations, though it is unclear if the phoneme-based MT model has an advantage over the BPE-based MT model. We follow the tradition of using the latter one and leave the comparison of them in future works.

Ethics Statement

WACO has the potential to benefit speakers of low-resource languages. For example, their published video or speech can be better translated into other languages, so more viewers in the world can understand them, enabling deeper communication between different cultures. Though WACO may be beneficial to cross-language communication, we do not encourage users to treat the translation generated by the E2E ST model as fully correct since they are far from perfect in practice.

Acknowledgement

References

Ashkan Alinejad and Anoop Sarkar. 2020. Effectively pretraining a speech translation decoder with machine translation data. In *Proc. of EMNLP*, pages 8014–8020.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, Dublin, Ireland. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben

Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proc. of NAACL-HLT*, pages 58–68.

Ankur Bapna, Yu-an Chung, Nan Wu, Anmol Gulati, Ye Jia, Jonathan H. Clark, Melvin Johnson, Jason Riesa, Alexis Conneau, and Yu Zhang. 2021. Slam: A unified encoder for speech and language modeling via speech-text joint pre-training.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 6224–6228. IEEE Press.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 6224–6228.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. 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*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech and Language*, 66:101155.

Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Yves Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzm'an, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proc. of NAACL-HLT*, pages 2012–2017.

Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021a. Consecutive decoding for speech-to-text translation. In *Proc. of AAAI*.

Qianqian Dong, Rong Ye, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021b. Listen, understand and translate: Triple supervision decouples end-to-end speech-to-text translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12749–12759.

Yichao Du, Weizhi Wang, Zhirui Zhang, Boxing Chen, Tong Xu, Jun Xie, and Enhong Chen. 2022. Non-parametric domain adaptation for end-to-end speech translation.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. 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*, pages 949–959, San Diego, California. Association for Computational Linguistics.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with speech-text manifold mixup for speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland. Association for Computational Linguistics.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: FBK@IWSLT2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence

data with recurrent neural networks. In *Proc. of ICML*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proc. of EMNLP*, pages 1317–1327.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP*, pages 66–71.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2022. Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 245–254, Dublin, Ireland. Association for Computational Linguistics.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the modality gap for speech-to-text translation. *ArXiv preprint*, abs/2010.14920.

Arya D. McCarthy, Liezl Puzon, and Juan Pino. 2020. Skinaugment: Auto-encoding speaker conversions for automatic speech translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7924–7928.

Chenggang Mi, Lei Xie, and Yanning Zhang. 2022. Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing. *Neural Netw.*, 148(C):194–205.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics,*

Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015, pages 5206–5210.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Proc. of INTERSPEECH*, pages 2613–2617.

Juan Miguel Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-training for end-to-end speech translation. In *Proc. of INTERSPEECH*, pages 1476–1480.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Michael Saxon, Samridhi Choudhary, Joseph P. McKenna, and Athanasios Mouchtaris. 2021. End-to-End Spoken Language Understanding for Generalized Voice Assistants. In *Proc. Interspeech 2021*, pages 4738–4742.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. 2022. Unified speech-text pre-training for speech translation and recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1488–1499, Dublin, Ireland. Association for Computational Linguistics.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. Improving speech translation by understanding and learning from the auxiliary text translation task. 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)*, pages 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A general multi-task learning framework to leverage text data for speech to text tasks. In *Proc. of ICASSP*, pages 6209–6213. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. Covost 2 and massively multilingual speech translation. In *Proc. Interspeech 2021*, pages 2247–2251.

Chen Wang, Yuchen Liu, Boxing Chen, Jiajun Zhang, Wei Luo, Zhongqiang Huang, and Chengqing Zong. 2022. Discrete cross-modal alignment enables zero-shot speech translation.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation. In *Proc. of ACL*, pages 3728–3738.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Proc. Interspeech 2017*, pages 2625–2629.

Anne Wu, Changhan Wang, Juan Miguel Pino, and Jiatao Gu. 2020. Self-supervised representations improve end-to-end speech translation. In *Proc. of INTERSPEECH*, pages 1491–1495.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proc. of ACL*, pages 2619–2630.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Proc. of INTERSPEECH*, pages 2267–2271.

Rong Ye, Mingxuan Wang, and Lei Li. 2022a. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Rong Ye, Mingxuan Wang, and Lei Li. 2022b. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022a. Revisiting end-to-end speech-to-text translation from scratch. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26193–26205. PMLR.

Ziqiang Zhang, Long Zhou, Junyi Ao, Shujie Liu, Lirong Dai, Jinyu Li, and Furu Wei. 2022b. Speechut: Bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training. In *Proc. of EMNLP*.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12736–12746.

A Appendix

A.1 Statistics of Datasets

We show statistics of MuST-C, IWSLT Mt-En, LibriSpeech and WMT datasets in Table 6,7,8 and 9.

Direction	Hours	# Sentence
En-De	408	234K
En-Fr	492	280K
En-Es	504	270K

Table 6: Statistics of MuST-C.

Type	Hours	# Sentence
En-De ST	1	0.6K
	10	5.8K
En ASR	10	5.8K
	100	58K
	370	216K
	1330	497K
Mt-En ST	1	0.9K
Mt ASR	10	6.7K

Table 7: Statistics of ST and ASR subsets in MuST-C En-De Low-Resource and IWSLT Mt-En.

Language	Hours	# Sentence	# Speaker
En	960	281K	2338

Table 8: Statistics of LibriSpeech.

Direction	Name	# Sentence
En-De	WMT16	4.6M
En-Fr	WMT14	40.8M
En-Es	WMT13	15.2M

Table 9: Statistics of WMT.

A.2 Impact of Hyperparameter

Temperature Figure 7 demonstrates BLEU scores produced by different temperature values

ST Data	1h			10h	
ASR Data	10h	100h	370h	100h	370h
Cascade w/o w2v2	<1	4.7	10.7	4.7	10.7
WACO w/o w2v2	**14.3**	**15.4**	**15.3**	**20.9**	**22.3**
Cascade w/ w2v2	**15.6**	**17.2**	**18.0**	17.2	18.0
WACO w/ w2v2	14.1	16.2	16.6	**21.0**	**22.7**

Table 10: Case-sensitive detokenized BLEU scores on MuST-C En-De `tst-COMMON` set of WACO and Cascade ST. We initializes the speech encoder with wav2vec 2.0 or random weights instead.

ST Data	1h				10h		
ASR Data	10h	100h	370h	1330h	100h	370h	1330h
WACO	45.5	40.6	42.7	33.3	24.9	21.7	21.3

Table 11: Word Error Rate (WER) of WACO on ASR part of MuST-C En-De `tst-COMMON` set.

in low-resource ST setting with 10h ASR and 1h ST data. Higher temperature in general leads to higher BLEU scores, but the marginal improvement becomes negligible when $\tau > 0.5$.
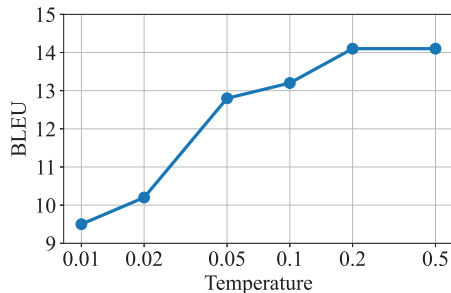

Figure 7: Case-sensitive detokenized BLEU scores on MuST-C En-De `tst-COMMON` set with different temperature values.

Pooling and Layer Here we compare different pooling mechanisms used to aggregate word-level representation and also different layers used to extract representations (before or after joint Transformer encoder). Table 12 shows BLEU scores in low-resource ST setting with 10h ASR and 1h ST data. WACO is not very sensitive to pooling method, but sensitive to the layer selection. Aligning representations extracted before joint encoder delivers much better performance.

A.3 More Examples of WACO versus ConST

We show two more examples that WACO achieves more accurate translation than ConST by better speech-text alignment in Figure 9.

Pooling Layer	Mean Before	Max Before	Sum Before	Mean After
BLEU	14.1	14.7	14.1	8.4

Table 12: Case-sensitive detokenized BLEU scores on MuST-C En-De `tst-COMMON` set with different pooling mechanisms (used to aggregate word-level feature) and layers (used to extract speech and text representations).

A.4 Loss Curves for Cross-Modal Pre-training

We present pre-training loss curves of CTC with both BPE and phoneme tokenizations, and WACO in Figure 8.

A.5 MT Pre-training

We use the same vocabulary and SentencePiece model described in Section 4.2 to tokenize the WMT datasets. The model is optimized with Adam. The learning rate starts at 1e-7, warmed up to 7e-4 by 4k steps and then decays following the inverse square root schedule with a minimum learning rate of 1e-9. The maximum number of tokens in a batch is 8192. We select the checkpoint with the highest BLEU (beam size 4, length penalty 0.6) on the WMT validation set.

A.6 Sequence-Level Knowledge Distillation and Cascade ST

SeqKD We apply the same MT model used to initialize joint Transformer in WACO to translate the transcript of ASR data into target language,

which is German in our case. The translation is produced by beam search with width 10 and length penalty 0.6. After we construct the pseudo ST dataset (10h), we combine it with the real-world ST dataset (1h) and obtain an 11h triplet ST dataset. Then we follow the same fine-tuning procedure in low-resource ST setting, i.e., cross-entropy losses, to obtain the final ST model.

Cascade ST We follow the same fine-tuning procedure used in (Baevski et al., 2020) to fine-tune wav2vec 2.0 small on 10h ASR data to obtain the ASR model. The final transcript is decoded by Viterbi algorithm. We use the same MT model and configuration as in SeqKD to translate the English transcript into German translation.

### A.7 Results of More Data Configurations

We compare WACO and Cascade ST with or without wav2vec2 in more data configurations as in Table 1 except for 1330-hour ASR. The results are shown in Table 10.

### A.8 Word Error Rate of using WACO as an ASR model

The word error rates (WER) of WACO with different data configurations are shown in Table 11.

A.9 Training Efficiency

The computation cost of calculating WACO loss function is higher than that of sentence-level methods (e.g., ConST) and in our profiling results, it takes 20~30% time of a forward pass. However, WACO converges much faster, in terms of number of iterations, than ConST and CTC. In the 100h ASR data case, WACO only needs <25k iterations to converge while both ConST and CTC requires >50k iterations. This makes WACO more time efficient overall.
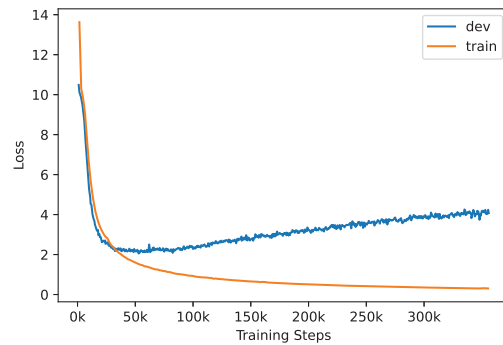
(a) CTC with BPE tokenization
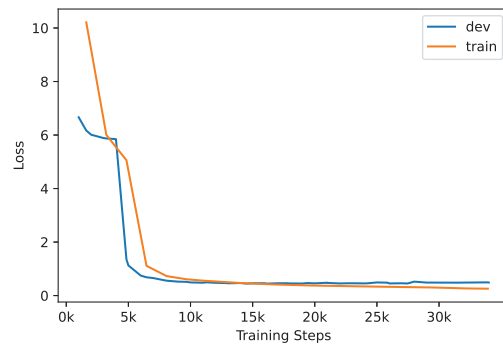
(b) CTC with phoneme tokenization
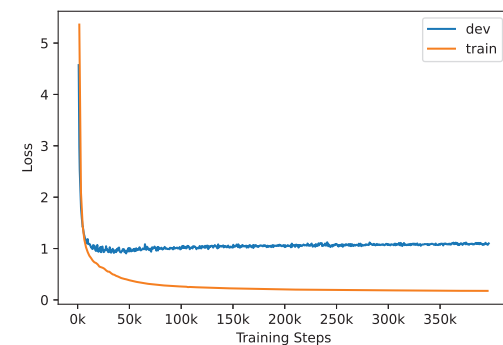
(c) WACO

Figure 8: Loss curves of various cross-modal pre-training method. CTC with BPE tokenization cannot generalize well to unseen speech (cannot reach below 2 on dev set).
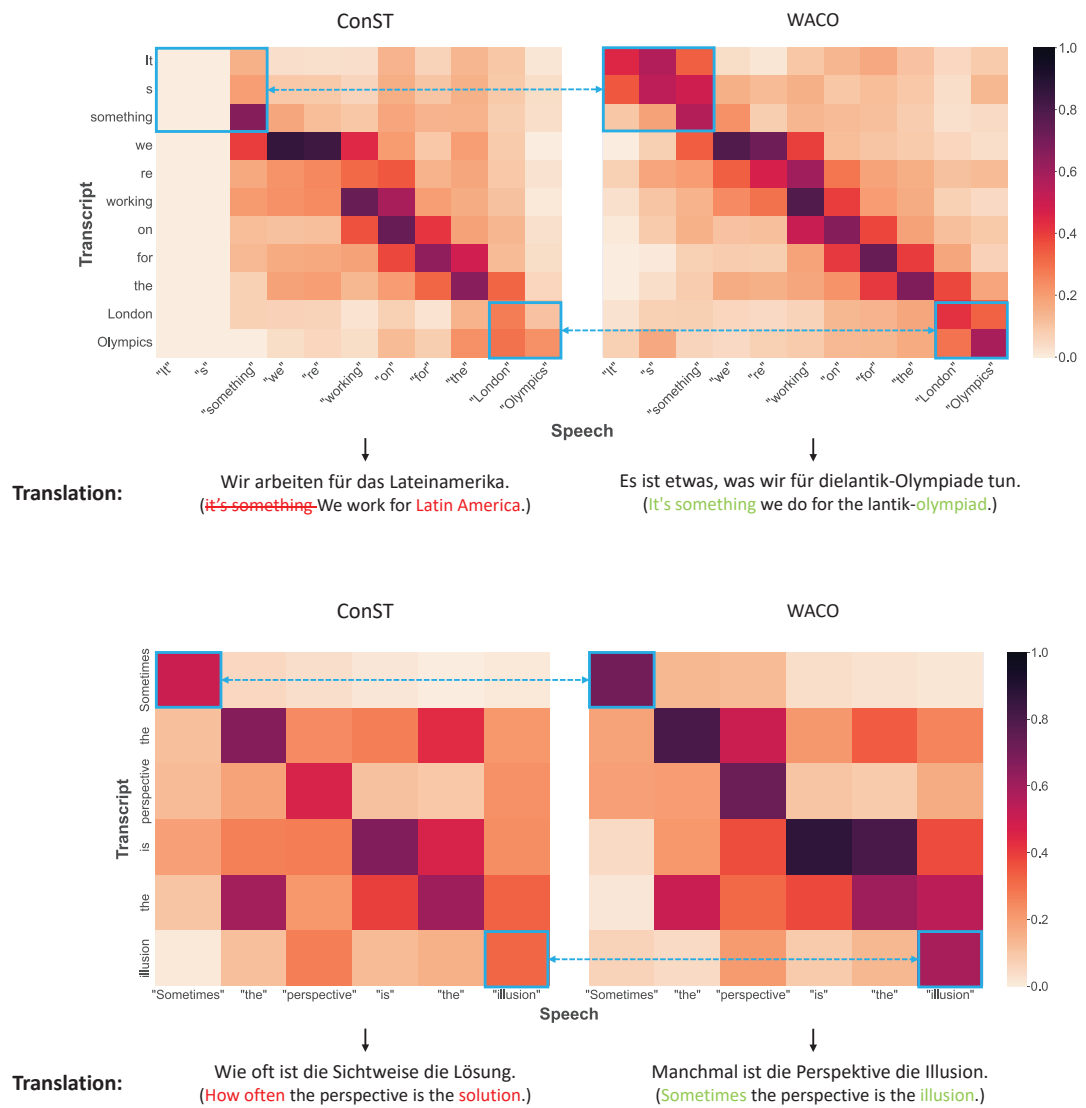
Figure 9: Two additional examples with speech-text alignment matrices and translations of WACO and ConST.

A   For every submission:

☑ A1. Did you describe the limitations of your work?
section 7

☑ A2. Did you discuss any potential risks of your work?
section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
section 0 and section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B   ☑ Did you use or create scientific artifacts?

section 4

☑ B1. Did you cite the creators of artifacts you used?
section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided
that it was specified? For the artifacts you create, do you specify intended use and whether that is
compatible with the original access conditions (in particular, derivatives of data accessed for research
purposes should not be used outside of research contexts)?
section 4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any
information that names or uniquely identifies individual people or offensive content, and the steps
taken to protect / anonymize it?
The data we use are all from public available and widely-used benchmarks.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and
linguistic phenomena, demographic groups represented, etc.?
section 4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits,
etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the
number of examples in train / validation / test splits, as these provide necessary context for a reader
to understand experimental results. For example, small differences in accuracy on large test sets may
be significant, while on small test sets they may not be.
We include the statistics in both section 4 and appendix.

C   ☑ Did you run computational experiments?

section 4 and 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget
(e.g., GPU hours), and computing infrastructure used?
section 4

---

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing
assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 4 and appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 4 and 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.