Joint Speech Transcription and Translation: Pseudo-Labeling with Out-of-Distribution Data

Mozhdeh Gheini[◇*], **Tatiana Likhomanenko**[†], **Matthias Sperber**[†], **Hendra Setiawan**[†]
[◇]Information Sciences Institute, University of Southern California
[†]Apple
gheini@isi.edu, {antares,sperber,hendra}@apple.com

Abstract

Self-training has been shown to be helpful in addressing data scarcity for many domains, including vision, speech, and language. Specifically, self-training, or pseudo-labeling, labels unsupervised data and adds that to the training pool. In this work, we investigate and use pseudo-labeling for a recently proposed novel setup: joint transcription and translation of speech, which suffers from an absence of sufficient parallel data resources. We show that under such data-deficient circumstances, the unlabeled data can significantly vary in domain from the supervised data, which results in pseudo-label quality degradation. We investigate two categories of remedies that require no additional supervision and target the domain mismatch: pseudo-label filtering and data augmentation. We show that pseudo-label analysis and processing in this way results in additional gains on top of the vanilla pseudo-labeling setup providing a total improvement of up to 0.4% absolute WER and 2.1 BLEU points for En–De and 0.6% absolute WER and 2.2 BLEU points for En–Zh.

1 Introduction

Semi-supervised learning methods have been a cornerstone in addressing annotated data scarcity by taking advantage of and incorporating the relatively larger amounts of *unlabeled*[1] data in the training process. Self-training is a relatively early instance of such methods (Scudder, 1965). Conceptually, self-training is simple: first, a base model is trained using limited labeled data. The base model is then used to predict labels for the unlabeled data. The generated labels are termed "*pseudo-labels*" (PLs) to signify their predicted nature, as opposed to gold supervised data. Finally, the pseudo-labels are combined with the initial seed supervised data to train a new model, and this process is repeated until no further improvement in performance is observed.

Self-training, or pseudo-labeling interchangeably, has been shown to be effective to improve upon fully supervised baselines in low-resource settings for several sequence-to-sequence (seq2seq) tasks, such as machine translation (MT) (Zhang et al., 2018; He et al., 2020; Jiao et al., 2021), end-to-end speech recognition (ASR) (Xu et al., 2020; Park et al., 2020; Kahn et al., 2020; Likhomanenko et al., 2021), end-to-end speech translation (ST) (Pino et al., 2020), and more recently speech-to-speech translation (Dong et al., 2022). In this work, we study pseudo-labeling for a recently proposed new setup, joint speech transcription and translation (STT) (Anastasopoulos and Chiang, 2018; Sperber et al., 2020): a setup that is of interest in use cases where both the transcript and translation of a speech signal are returned to the user. As we describe in detail later in §2.1, the fully supervised data for modeling end-to-end joint transcription and translation is triples of form (s, tc, tl) where s is the speech signal, tc is the transcript, and tl is the translation. As that is especially costly to come by, STT also seems to have the potential to benefit from pseudo-labeling.

Our investigations show that while pseudo-labeling (PL) is indeed helpful, the quality of pseudo-labels that bring about the benefits is subpar. Upon inspecting the supervised and unsupervised sets, that proves to be not surprising: with limited amounts of supervised data, it is likely that the supervised and unsupervised sets differ in domain, impacting the quality of pseudo-labels. Specifically, in our case, we identify two causes leading to domain mismatch with out-of-distribution unlabeled data: difference between the sequence length ranges and vocabulary sets of the supervised and unsupervised sets. In this work, we ask *if* we can specifically counteract the domain mismatch to reach a set of pseudo-labels of higher quality,

[*] Work done during an internship at Apple.

[1] We use descriptors "(un)labeled" and "(un)supervised" interchangeably throughout this paper.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7637–7650
July 9-14, 2023 ©2023 Association for Computational Linguistics

and *if* that higher quality, in turn, translates into a better overall performance of pseudo-labeling.

First, we propose PLs filtering based on simple data-centric criteria inspired by Likhomanenko et al. (2021). While PLs filtering is a common component of PL algorithms, it is usually based on the model prediction scores (Kahn et al., 2020; Park et al., 2020; Zhang et al., 2021, 2022), which may not directly target the identified domain mismatch aspects, e.g., different sequence length ranges, as our proposed filtering does. Second, we propose augmenting the supervised data by concatenating randomly-picked samples to create new ones and adding them to the supervised set. These two are essentially different in nature: while filtering increases the overall quality by removing samples with PLs that are likely to be faulty, augmentation does so by extending the supervised set and generating better labels in the first place. Our results confirm that indeed this distinction in nature gets reflected in different ways filtering and augmentation improve the performance of pseudo-labeling.

The outline of this paper is as follows. We provide some background in §2 and detail the experimental setup in §3. Then, in §4, we report and discuss the results from vanilla pseudo-labeling, the observation of domain mismatch, and the gains brought about by filtering and augmentation.

Our **contributions** are: 1) We specifically focus on PL in the face of domain mismatch between the supervised and unsupervised sets; 2) We investigate the mitigation of the effect of domain mismatch through two approaches: PLs filtering and augmentation by concatenation and demonstrate how they improve PL in different ways. These approaches can be repurposed wherever PL is considered as a solution; 3) We apply PL modified with those approaches specifically to a novel setup, joint speech transcription and translation, and report gains on top of the vanilla PL for STT.

2 Background

Our work studies a pseudo-labeling solution for end-to-end joint speech transcription and translation. In this section, we provide the background for these two components involved in the study, namely *speech transcription and translation* and *pseudo-labeling*.

2.1 Speech Transcription and Translation

Our task of speech transcription and translation (STT) is closely related to script recognition (ASR) and speech translation (ST). ASR is the task of generating the text equivalent to an audio speech signal. Meanwhile, ST aims to generate the text equivalent to the signal in a target language other than the language of the speaker. In contrast, STT generates both the transcript and the translation *jointly* in an end-to-end fashion. STT is particularly appealing in cases where both the transcript and translation are to be displayed to the user.

Formally, STT can be modeled as follows: given a speech signal (s), the model generates the transcript (tc) and translation (tl) concatenated together in the output as one single sequence: $s \rightarrow tc\_tl$ (Sperber et al., 2020). This formulation is simple to implement as it casts STT as an instance of the well-known seq2seq modeling and results in a *single* end-to-end model to be stored on device. Furthermore, as reported by Sperber et al. (2020), this formulation results in a reasonably consistent transcripts and translations as the coupled inference ensures that translations are conditioned on the transcripts. In our experiments, we use this STT formulation as it offers a good trade-off between accuracy, computational efficiency, and consistency.

However, the major challenge that such modeling presents is insufficient data resources: three-way parallel samples of form (s, tc, tl) are expensive to annotate. Annotation would require multilingual annotators and would be time-consuming. To alleviate this limitation, we study how pseudo-labeling can be employed effectively to combat data scarcity in this setting. We provide a background on pseudo-labeling in the next section.

2.2 Pseudo-labeling

Pseudo-labeling (PL), often referred to as self-training in the literature, addresses the data insufficiency issue by taking advantage of much larger amounts of unsupervised data. More precisely, assume a labeled set $L = \{x_i, y_i\}$ and an unlabeled set $U = \{x_j\}$, where $|U| \geq |L|$, are available (note that in the case of STT, y_i is actually a tuple consisting of the transcript and the translation: $y_i = (tc_i, tl_i)$). PL starts with training an initial model M in a supervised manner using L. Then, using M, it generates pseudo-labels (predictions) for U. It then incorporates the pseudo-labels (PLs) to create a new model M^+, which hopefully super-

Algorithm 1 Pseudo-labeling

```
Require: $L = \{x_i, y_i\}$ and $U = \{x_j\}$
Train a base model $M$ on $L$
while The desired number of rounds or convergence has not been reached do
    Generate the pseudo-labeled set: $P = \{x_j, M(x_j) \mid x_j \in U\}$
    Obtain $M^+$ by fine-tuning $M$ on $L \cup P$
    Replace $M$ with $M^+$
end while
return $M$
```

sedes M in performance. M^+ can then replace M to repeat this process for as many rounds as desired, or until no further gains are observed. Although conceptually simple, several key decisions need to be made before PL can be applied.

How should M^+ be created? M^+ can be trained from scratch (Park et al., 2020) or alternatively obtained by continuously fine-tuning M (Xu et al., 2020) using the labeled set combined with the pseudo-labeled set. As we later report in §4, in our experiments, fine-tuning consistently outperforms training from scratch. Hence, we opt for fine-tuning in our experiments.

Should PL be applied to supervised set? For the PL stage, we consider and experiment with labeling the supervised set in addition to the unsupervised set and monitor for any potential improvements. Similar to the previous item, as we later show in §4, using PLs for the supervised set does not prove to be beneficial in our experiments. Therefore, we generate predictions only for the unlabeled set.

In what way should the pseudo-labels be used to update existing models? For instance, He et al. (2020), at each round, first train a model from scratch on the pseudo-labeled set, and then fine-tune it on the supervised set to obtain the final model for that round. Alternatively, Xu et al. (2020) combine the two sets and use a hyper-parameter to have control over the relative weight of the supervised portion against the pseudo-labeled portion. To keep our setup simple, we opt for combining the sets and treating them equally.

With the key factors outlined above, Algorithm 1 shows how we carry out vanilla pseudo-labeling for our experiments. All results we report in §4.1 follow this algorithm.

3 Experimental Setup

3.1 Data

In this work, we use two publicly available multilingual speech translation datasets which, thanks to the nature of their creation, include transcripts: CoVoST V2 (Wang et al., 2020) and MuST-C (Cattoni et al., 2021). CoVoST V2 is created by amending the validated audio clips and transcripts from the Common Voice crowd-sourced ASR corpus (Ardila et al., 2020) with professional translations. It covers translations from English into 15 languages and from 21 languages into English. MuST-C is created by automatically aligning the audio segments from TED talks to corresponding manual transcripts and translations (available from the TED website), which are also aligned. It covers translations from English into 14 languages.

We conduct our experiments across two language pairs: English–German (En–De) and English–Chinese (En–Zh), which are available in both CoVoST and MuST-C. In all our experiments, we designate CoVoST as the supervised set, and MuST-C as the unsupervised set. Note that this means our objective is to reach the best performance possible on the CoVoST evaluation set. While we also have the gold transcripts and translations (labels in the STT problem) for MuST-C, we do not use them and practically treat MuST-C as an unlabeled set. We only use MuST-C gold labels for analysis and pseudo-label quality assessment. We provide the statistics of our data in Table 1.

3.2 Model

To extract speech representations, we first use pre-trained wav2vec 2.0 Base (Baevski et al., 2020)[2] which results in 20ms per frame. On top of this extractor, we use a stack of three convolutional lay-

[2] We use a model provided by Hugging Face Transformers (Wolf et al., 2020): `facebook/wav2vec2-base-960h`.

	CoVoST		MuST-C	
	Train	Eval	Train	Eval
En–De	233k	15.5k	251k	1.4k
En–Zh	233k	15.5k	359k	1.3k

Table 1: Amount of data available (number of sentences), per language pair and corpus.

ers to downsample the input further, resulting in 160ms per frame: each layer has a kernel of 3 and a stride of 2. Next we attach encoder-decoder Transformer (Vaswani et al., 2017) with pre-layer normalization, a hidden dimension of 1024, dropout of 0.1, and five and three layers of encoder and decoder, respectively, following Sperber et al. (2020). Positional embeddings (absolute sinusoidal) are only added on the decoder side. The whole model is trained in an end-to-end manner, including the wav2vec 2.0 feature extractor. On the output side, as described in §2.1, the decoder generates one sequence consisting of the transcript and the translation concatenated together.

In terms of input prepossessing, we remove instances where speech is either shorter than 0.5s or longer than 15s, or either the transcript or the translation is longer than 50 words. After that, we use `SentencePiece` (Kudo and Richardson, 2018) for subword tokenization. The vocabulary is created using only the supervised set. We use a vocabulary size of 1020 and 8188 in the case of En–De and En–Zh, respectively. The transcription and translation vocabulary is shared in both cases.

The objective function during optimization is a weighted sum of the CTC loss (Graves et al., 2006) on the encoder side and the cross-entropy loss on the decoder side. For both training a base model and fine-tuning an existing checkpoint on the union of the labeled set and the pseudo-labeled set, we use Adam optimizer (Kingma and Ba, 2015) with peak learning rate of 0.0005 after 500 warmup steps, coupled with inverse square root learning rate scheduling. We train for a total of 100 epochs and use SpecAugment (Park et al., 2019) in the same way and with the same parameters as wav2vec 2.0. After training, pseudo-labels are generated with a beam size of five.

For both language pairs, we use the dev sets provided by the corpora as the held-out evaluation set. For scoring (and only for scoring), we remove diacritics and punctuation, and report our performance

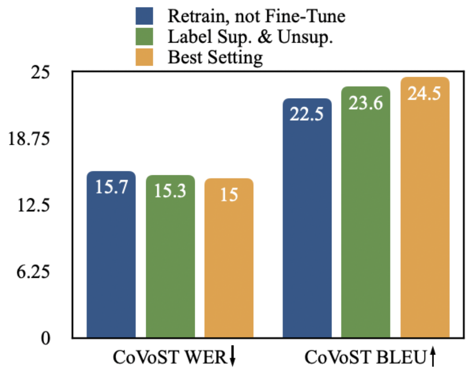

Figure 1: Performance of different PL settings on our supervised set, CoVoST. In each case, PL is done for three rounds. The best setting fine-tunes the checkpoint from the last round on the supervised set and the pseudo-labels for the unsupervised set.

in terms of word error rate (WER) of transcripts and BLEU of translations using beam size of five with SacreBLEU.[3]

Our implementation is built upon `PyTorch` (Paszke et al., 2019), `xnmt` (Neubig et al., 2018), and `Lightning` (Falcon and The PyTorch Lightning team, 2019).

4 Results and Discussion

We present our results in this section in the following order: §4.1 establishes vanilla pseudo-labeling performance, which leads to our analysis of the domain mismatch between the supervised and unsupervised sets. §4.2 and §4.3 then describe the two categories of remedies we devise to mitigate the effect of domain discrepancies on pseudo-labeling.

As mentioned in §2.2, this is all using the best setting we were able to establish during our pilot experiments: at each pseudo-labeling round, we 1) label only the unsupervised data, and 2) fine-tune the existing checkpoint on the combination of supervised and pseudo-labeled data. We conduct our pilot experiments on En–De. We were able to confirm that the aforementioned setting consistently beats the rest over several rounds of pseudo-labeling. Figure 1 illustrates the lead of the best setting over others in the last round of our experiments. The same pattern holds across all rounds.

4.1 Vanilla Pseudo-Labeling

In Table 2, we include the results of vanilla PL, as in Algorithm 1, with no modifications. We report

[3] Hash: case.lc+numrefs.1+smooth.4.0+tok.{13a,zh} for {En–De,En–Zh}.

		En–De					En–Zh					
		Base Model	R1	R2	R3	Bound	Base Model	R1	R2	R3	R4	Bound
🔍CoVoST	WER ↓	15.4	15.4	15.0	15.0	14.4	14.8	14.6	14.8	14.7	14.6	13.7
	BLEU ↑	22.8	23.8	24.5	24.5	25.5	28.7	29.4	30.0	30.5	30.7	31.9
MuST-C	WER ↓	45.1	45.2	29.7	28.4	9.6	47.9	46.2	43.8	42.8	37.2	8.9
	BLEU ↑	7.3	9.1	9.7	9.6	22.4	9.1	9.9	9.6	9.0	8.3	18.9

Table 2: Vanilla pseudo-labeling results over each round up to saturation. CoVoST, our supervised set, is distinguished with 🔍symbol to signify it is intended to improve performance on it. MuST-C is the unsupervised set. “Bound” refers to the performance of fully supervised models trained on the combination of CoVoST and MuST-C with their gold labels.

WER and BLEU for En–De and En–Zh across both corpora. To reiterate, CoVoST (distinguished by the magnifying glass symbol 🔍) is our designated supervised set, and hence, what we are trying to boost performance on. MuST-C scores, on the other hand, are reported for the sake of analysis; the metrics are to assess the quality of PLs.

We report the performance of the initial model (the fully supervised baseline, Model M on line 1 of the Algorithm 1) in the “Base Model” column. Scores from each pseudo-labeling round, thereafter, appear on the corresponding “R” column. To have an upper bound of what is possible with the collective data *if* pseudo-labels were predicted perfectly, we train a single model using both corpora in a supervised manner. Those numbers are provided in the “Bound” column. Note that this is the only case for which MuST-C gold labels are used.

First and foremost, in confirmation with the literature, vanilla pseudo-labeling is effective. On 🔍CoVoST, it is able to improve the base model by 0.4% absolute WER and 1.7 BLEU points on En–De, and 0.2% absolute WER and 2.0 BLEU points on En–Zh. However, with a closer look at the quality of pseudo-labels at each round (i.e., MuST-C scores), it is evident that the generated labels are far from ideal quality.

Our investigation into the reasons as to why that is the case points to two root causes that indicate 🔍CoVoST and MuST-C are significantly different in *domain* in the following aspects:

Length mismatch between corpora. As shown in Figure 2, MuST-C speech sequences are generally longer, which also results in longer transcripts and translations.

Vocabulary mismatch between corpora. We were also able to identify discrepancies between the vocabulary of words between the two corpora.

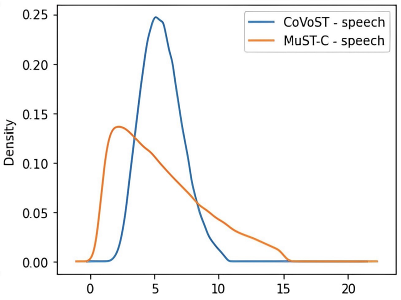

Figure 2: The PDF of input audio lengths (in seconds) estimated using kernel density estimation. MuST-C speech signals are longer in duration.

For instance, on the English side, MuST-C and CoVoST each have roughly 64k and 121k unique types, respectively. Of those, only 38k types are in common, with CoVoST having more probability mass on rare (tail-end of the Zipfian distribution) vocabulary types. Specifically, even if we train plain machine translation systems on 🔍CoVoST transcripts and translations (and take the audio out of the picture), the En–De system scores only 12.4 BLEU on MuST-C En–De, and the En–Zh system scores only 9.6 BLEU on MuST-C En–Zh.

Following these observations, we next demonstrate that it is possible to counteract the domain mismatch and enhance the quality of labels to boost the effectiveness of pseudo-labeling.

4.2 Direction #1: Data-Centric Filtering

Per §2.2, in vanilla PL, we use all the generated labels to update the model. Alternatively, PLs can be filtered to remove predictions of less quality. Recent works (Park et al., 2020) rely on confidence scores from the model to filter the pseudo-labels, which require careful and proper normalization. Kahn et al. (2020) use a combination of heuristic-based and confidence-based filtering. In our case, similar to Likhomanenko et al. (2021), we propose

	En–De				En–Zh			
	🔍CoVoST		MuST-C		🔍CoVoST		MuST-C	
	WER ↓	BLEU ↑	WER ↓	BLEU ↑	WER ↓	BLEU ↑	WER ↓	BLEU ↑
Bound	14.4	25.5			13.7	31.9		
Vanilla PL	15.4/**15.0**	23.8/24.5	45.2/28.4	9.1/9.7	14.6/14.6	29.4/30.7	46.2/37.2	9.9/9.9
Ratio to Gold	15.3/15.0	24.1/24.7	22.8/15.8	9.6/10.4	14.5/14.2	29.5/30.5	23.2/17.4	10.0/10.2
Ratio KDE	**15.1/15.0**	24.2/24.5	30.5/27.1	9.4/10.1	**14.3/14.2**	29.8/30.7	30.8/21.7	10.8/10.8
LASER	15.2/**15.0**	24.1/24.5	34.7/27.6	9.6/10.0	14.6/14.3	29.4/30.6	40.8/20.3	10.7/11.2
Augmentation	15.3/15.3	**24.9/24.9**	33.8/22.2	11.5/11.8	14.6/14.3	**30.1/30.9**	48.7/25.4	11.9/11.9

Table 3: Improved results using remedies recommended. Each cell includes the performance obtained from the first round and the best performance obtained using the corresponding method (R1/Best). We also include bounds from Table 2 for 🔍CoVoST for comparison. We use bold font to mark the best performance on 🔍CoVoST.

and only rely on data-centric metrics to specifically target domain-mismatch and select a subset of pseudo-labels to use in the next round: transcript length to audio length ratio and transcript and translation LASER embeddings cosine similarity.

4.2.1 Length Ratio Distribution

A sign of flawed inference and faulty output in seq2seq models has been known to be looping (Chorowski and Jaitly, 2017): the model generates the same n-gram repeatedly. We were also able to identify looping occurring frequently in the PLs and resulting in long transcripts. While the supposed lengths of the correct transcripts are unknown, the length of the input audio can be used as an indicator: heuristically, the shorter the input audio, the shorter the transcript.

To take advantage of this signal with no supervision overhead, we estimate the probability density function (PDF) of the joint probability distribution over the input audio lengths and predicted transcripts lengths using kernel density estimation (KDE). At each PL round then, we only keep the top 90% (found empirically) of the most probable transcripts. Figure 3 visualizes the effect of such filtering. Instances that have the highest PDF values, have a similar ratio of transcript length to audio length to that of gold transcripts. Hence, this can be a useful metric that needs no additional supervision.

To gauge the maximum potential effectiveness of length ratio-based filtering, we also conduct experiments with filtering based on the ratio of the generated transcript length to the *gold* transcript length, where we only keep those with the length within 0.9× and 1.1× the length of the corresponding gold transcript. Note that this only has discussion purposes, as it uses supervision in the form of access to the length of the gold transcripts.

Table 3 (rows "Ratio to Gold" and "Ratio KDE") shows how our length ratio-based filtering methods compare against plain vanilla pseudo-labeling. For each method, we run the same number of rounds as we did for vanilla pseudo-labeling in Table 2. We report the performance of the first round and the best round (first round/best round in table cells) of each method. Results from each separate round are comprehensively provided in Appendix A.

On 🔍CoVoST, "Ratio KDE" speeds up gains relative to vanilla pseudo-labeling despite incorporating fewer labels (only 90%): 15.1 vs. 15.4 WER and 24.2 vs. 23.8 BLEU at the first round in the case of En–De. The same pattern holds for En–Zh. Looking at the scores on MuST-C, it is evident that moderating the quality of pseudo-labels in this way, does indeed translate into better pseudo-labels for future rounds and improved performance on the supervised set. Also, "Ratio to Gold", benefiting from a form of supervision, expectedly results in better quality on the unsupervised set. However, on the supervised set, it performs similarly to "Ratio KDE", demonstrating that "Ratio KDE" is effective enough at removing detrimental pseudo-labels.

While "Ratio KDE" performs clearly better at earlier rounds, it saturates at the same performance as vanilla pseudo-labeling, which uses all the labels (with being better only in the case of En–Zh WER by 0.4% absolute WER). So it is especially beneficial when available resources can only cover a small number of pseudo-labeling rounds.

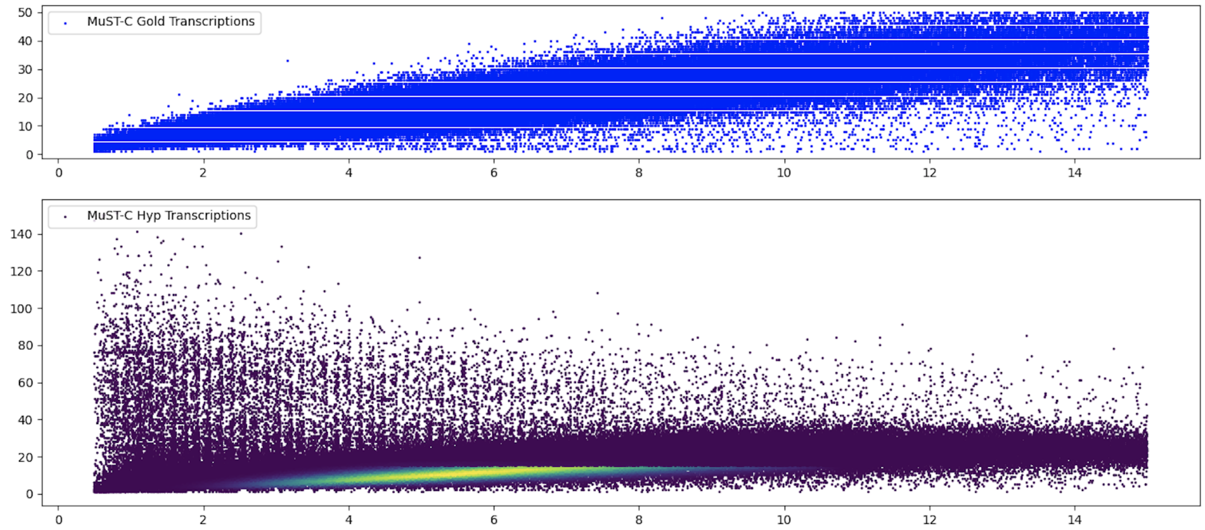

Figure 3: Plots of transcript lengths (y-axis, length in words) against input audio lengths (x-axis, length in seconds) for gold transcripts (top) and generated transcripts during pseudo-labeling (bottom). Datapoints in the bottom plot are color-coded based on their PDF value as estimated by KDE, with lighter colors indicating higher values. The most probable mass forms a pattern similar to that of gold transcription. Therefore PDF values can be effective for filtering pseudo-labels of less quality.

4.2.2 LASER Score

Our second filtering method relies on the relationship between the generated translations and transcripts (this is in contrast to the previous method, which relied on the relationship between the generated transcripts and audio signals). For this, we use the pretrained LASER model (Artetxe and Schwenk, 2019), a multilingual sentence encoder, to embed the generated transcripts and translations in a multilingual space to rank pairs based on the cosine similarity and hold onto only the top 90%. Given that LASER lies at the center of this, the quality of representations of different languages in its multilingual space can affect the degree of gains it can bring about.

Per Table 3, row "LASER", LASER-based filtering improves performance on the unsupervised set (and hence, the quality of the PLs) all across the board. Those improvements translate into better performance on the supervised set for both En–De and En–Zh. Importantly, the improvement pattern is similar to that of length ratio-based filtering: more gains at earlier rounds, saturating at the same performance as the vanilla PL. However, as opposed to ratio-based filtering, which needs no additional supervision, the LASER model is trained using a massive amount of bitext and benefits from supervision in that way. But that does not result in enhanced performance compared to ratio-based filtering. So while LASER scores present a second avenue for pseudo-label filtering, "Ratio KDE" incurs strictly no supervision overhead, is simple, and is the best-performing filtering method.

4.3 Direction #2: Data Augmentation

Our previous filtering methods remove PLs so that the remaining subset has a higher quality. However, if we can generate better labels, to begin with, we can discard none and retain all the labels. Here, to improve the quality of the labels generated by the base model at no extra supervision cost, we use data augmentation by concatenation to directly target the reported length mismatch between corpora in §4.1. To do so, we create an augmented set from our supervised set by randomly selecting a pair of samples and constructing a new sample by concatenating the audio signals as the input and concatenating corresponding transcripts and translations as output. In our experiments, we build a set of 20k augmented samples as such using the original 🔍CoVoST data. After training the base model, before generating PLs, we first further fine-tune the base model on the union of the original supervised set and the augmented set. We then proceed as in vanilla PL with the union of the original data and the augmented set as our supervised training set.

As shown in Table 3, row "Augmentation", although no generated labels are thrown away, the quality of PLs is indeed increased in the subsequent round. This is especially pronounced in the case

Ref. Transcription	It means reduce your carbon dioxide emissions with the full range of choices that you make, and then purchase or acquire offsets for the remainder that you have not completely reduced.
Ratio KDE	It means reduce your carbon dioxide emissions, with the full range of choices that you make, and then purchase or purchase or purchase.
Augmentation	It means reduce your carbon dioxide emissions. With the full range of choices that you make. And then purchase or acquire offsets for the remainder that you have not completely reduced.

Table 4: Pseudo-labels generated by "Ratio KDE" and "Augmentation". The reference is also provided. The label in the case of "Ratio KDE" gets filtered. But "Augmentation" gets to learn from it in the next round.

of translations. We provide an example evidencing this in Table 4. Here we compare the PLs generated by "Ratio KDE" and "Augmentation" for an utterance in MuST-C against each other. For a longer input, "Ratio KDE" suffers from looping and inadequate generation, and this instance actually gets filtered. However, "Augmentation" gets it right and retains it for training in the subsequent round. The fact that it also generates the output as sentences separated with periods indicates that this is indeed learned as a consequence of augmented samples.

With retaining all pseudo-labels, not only does bootstrapping the supervised set using concatenation expedite the gains from pseudo-labeling, but it is also the most effective in terms of the final performance before saturation by improving the score in three cases: it improves the performance of vanilla pseudo-labeling on 🔍CoVoST by 0.4 and 0.2 BLEU points on En–De and En–Zh, respectively, and by 0.3% absolute WER on En–Zh. Therefore, it further closes the gap between pseudo-labeling and the upper bounds.

To conclude our discussion on how domain mismatch can be addressed, we find filtering methods, which discard labels, to be only effective when due to any resource limitation, only a few rounds of pseudo-labeling can be run. This finding also echoes insights from Bansal et al. (2022) that studies data scaling laws for MT and shows while filtering may benefit computational efficiency, more unfiltered data can replace filtered data. As an alternative to filtering, we show that improving the quality of all generated labels through augmentation so that all can be kept, is the most effective, especially when as many rounds as needed can be run to reach saturation.

5 Related Work

The two paradigms often considered in low-resource data scenarios are self-training and pre-training. Self-training, or pseudo-labeling, has long been studied for a variety of seq2seq tasks (He et al., 2020; Xu et al., 2020; Park et al., 2020; Kahn et al., 2020; Chen et al., 2020; Likhomanenko et al., 2021; Pino et al., 2020; Dong et al., 2022). Regarding the relationship between pretraining and self-training, Xu et al. (2021) and Wang et al. (2021) show that self-training and unsupervised pretraining are complimentary and can be combined to boost performance on speech recognition and speech translation, respectively. In the case of supervised pretraining, however, Zoph et al. (2020) show in the vision domain that as the size of the labeled data available grows, self-training remains helpful, whereas the benefits of supervised pretraining start to diminish.

For applying self-training to the unvisited setup of joint speech transcription and translation (Sperber et al., 2020), we focus on domain mismatch, a matter which can get overlooked when gains from vanilla pseudo-labeling are observed. As solutions, we study pseudo-label filtering and augmentation by concatenation. In contrast to conventional filtering, which relies on normalized model confidence scores (Park et al., 2020; Kahn et al., 2020), or recently, the agreement between several forward passes of the model run with dropout (Khurana et al., 2021), we define and use data-centric factors that are attuned to the domain differences we observe and directly target them.

Concatenation as an effective augmentation method has been studied in the context of machine translation (Agrawal et al., 2018; Kondo et al., 2021; Nguyen et al., 2021; Gowda et al., 2022) and speech-to-text (Lam et al., 2022). In our case, we use it to expose our base model to sequences of higher length to improve the quality of generated pseudo-labels.

6 Conclusion

We study pseudo-labeling for joint speech transcription and translation. We show that while vanilla pseudo-labeling is helpful, additional improvements are obtained by addressing the low quality of generated pseudo-labels due to domain mismatch between the supervised and unsupervised sets.

We find that our proposed solutions help in two different ways, as they are in distinct nature: pseudo-label filtering, which discards low-quality labels, is mostly helpful by expediting gains in earlier rounds, especially for transcriptions. Augmentation by concatenation, on the other hand, does not discard any of the labels. As a result, it is able to maintain an edge over vanilla pseudo-labeling in the late rounds as well.

Limitations

We would like to acknowledge the following limitations of this work.

Our study setup only takes advantage of supervised data in the form of triples of <speech, transcriptions, translations>. This is because we first and foremost want to investigate the effectiveness of pseudo-labeling in the most extreme case. However, the setup can be extended to be able to also rely on ASR-only (<speech, transcription>) and ST-only (<speech, translation>) pairs. We leave incorporating ASR and ST data as a future work as well as incorporating external language and machine translation models.

We identified two sources of domain mismatch: input length ranges and vocabulary mismatch. However, the solutions that we investigate directly target the length mismatch, without explicitly addressing the vocabulary mismatch. The latter is indeed more challenging to address, especially without incurring additional supervision. In fact, circling back to the previous item as a future direction, incorporating supervision in the form of ASR or ST can expand the vocabulary set, also addressing vocabulary mismatch.

Acknowledgements

We would like to thank Qin Gao, Amittai Axelrod, Boliang Zhang, Barry Theobald, David Grangier, Jiatao Gu and the rest of machine translation and machine learning research teammates for fruitful discussions and constructive feedback on the manuscript.

References

Ruchit Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 11–20.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 82–91.

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 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

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*, volume 33, pages 12449–12460. Curran Associates, Inc.

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. 2022. Data scaling laws in NMT: The effect of noise and architecture. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1466–1482. PMLR.

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 & Language*, 66:101155.

Yang Chen, Weiran Wang, and Chao Wang. 2020. Semi-supervised ASR by end-to-end self-training. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2787–2791. ISCA.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 523–527. ISCA.

Qianqian Dong, Fengpeng Yue, Tom Ko, Mingxuan Wang, Qibing Bai, and Yu Zhang. 2022. Leveraging

Pseudo-labeled Data to Improve Direct Speech-to-Speech Translation. In *Proc. Interspeech 2022*, pages 1781–1785.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Thamme Gowda, Mozhdeh Gheini, and Jonathan May. 2022. Checks and strategies for enabling code-switched machine translation.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. 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 2840–2850.

Jacob Kahn, Ann Lee, and Awni Hannun. 2020. Self-training for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088.

Sameer Khurana, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2021. Unsupervised domain adaptation for speech recognition via uncertainty driven self-training. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6553–6557.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2022. Make more of your data: Minimal effort data augmentation for automatic speech recognition and translation. *CoRR*, abs/2210.15398.

Tatiana Likhomanenko, Qiantong Xu, Jacob Kahn, Gabriel Synnaeve, and Ronan Collobert. 2021. slimIPL: Language-Model-Free Iterative Pseudo-Labeling. In *Proc. Interspeech 2021*, pages 741–745.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: The eXtensible neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 185–192, Boston, MA. Association for Machine Translation in the Americas.

Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. Data augmentation by concatenation for low-resource translation: A mystery and a solution. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 287–293, Bangkok, Thailand (online). Association for Computational Linguistics.

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. *Proc. Interspeech 2019*, pages 2613–2617.

Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le. 2020. Improved Noisy Student Training for Automatic Speech Recognition. In *Proc. Interspeech 2020*, pages 2817–2821.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-Training for End-to-End Speech Translation. In *Proc. Interspeech 2020*, pages 1476–1480.

H. Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Matthias Sperber, Hendra Setiawan, Christian Gollan, Udhyakumar Nallasamy, and Matthias Paulik. 2020. Consistent transcription and translation of speech. *Transactions of the Association for Computational Linguistics*, 8:695–709.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus.

Changhan Wang, Anne Wu, Juan Pino, Alexei Baevski, Michael Auli, and Alexis Conneau. 2021. Large-Scale Self- and Semi-Supervised Learning for Speech Translation. In *Proc. Interspeech 2021*, pages 2242–2246.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. 2020. Iterative Pseudo-Labeling for Speech Recognition. In *Proc. Interspeech 2020*, pages 1006–1010.

Bowen Zhang, Songjun Cao, Xiaoming Zhang, Yike Zhang, Long Ma, and Takahiro Shinozaki. 2022. Censer: Curriculum semi-supervised learning for speech recognition based on self-supervised pre-training. *arXiv preprint arXiv:2206.08189*.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*, volume 33, pages 3833–3845. Curran Associates, Inc.

A Extended Results

	🔍CoVoST		MuST-C	
	WER ↓	BLEU ↑	WER ↓	BLEU ↑
Bound	14.4	25.5		
Base Model	15.4	22.8	45.1	7.3
Vanilla PL	15.4	23.8	45.2	9.1
	15.0	24.5	29.7	9.7
	15.0	24.5	28.4	9.6
Ratio to Gold	15.3	24.1	22.8	9.6
	15.0	24.5	18.5	10.2
	15.1	24.7	15.8	10.4
Ratio KDE	15.1	24.2	30.5	9.4
	15.0	24.5	27.7	9.8
	15.4	24.4	27.1	10.1
LASER	15.2	24.1	34.7	9.6
	15.0	24.5	29.1	9.9
	15.3	24.5	27.6	10.0
Augmentation	15.3	24.9	33.8	11.5
	15.3	24.9	22.2	11.8

Table 5: Extended results on En–De. All run until saturation. Each row represents one round of pseudo-labeling with the respective method.

	🔍CoVoST		MuST-C	
	WER ↓	BLEU ↑	WER ↓	BLEU ↑
Bound	13.7	31.9		
Base Model	14.8	28.7	47.9	9.1
Vanilla PL	14.6	29.4	46.2	9.9
	14.8	30.0	43.8	9.6
	14.7	30.5	42.8	9.0
	14.6	30.7	37.2	8.3
Ratio to Gold	14.5	29.5	23.2	10.0
	14.3	30.5	18.7	10.2
	14.4	30.5	17.9	9.7
	14.2	30.5	17.4	9.9
Ratio KDE	14.3	29.8	30.8	10.8
	14.3	30.2	22.0	10.8
	14.2	30.4	21.7	10.8
	14.2	30.7	21.7	10.4
LASER	14.6	29.4	40.8	10.7
	14.4	30.4	27.5	10.4
	14.4	30.5	24.4	10.4
	14.3	30.6	20.3	11.2
Augmentation	14.6	30.1	48.7	11.9
	14.5	30.5	35.7	11.0
	14.5	30.9	26.3	11.5
	14.3	30.9	25.4	11.3

Table 6: Extended results on En–Zh. All run until saturation. Each row represents one round of pseudo-labeling with the respective method.

B Responsible NLP Research

B.1 Computing Infrastructure

Our experiments are each run using 32 NVIDIA V100 GPUs (4 8-GPU nodes).

B.2 Licenses of Artifacts Used

We use the following artifacts in compliance with their terms of use:

- CoVoST V2 dataset (Wang et al., 2020) under CC BY-NC 4.0
- MuST-C dataset (Cattoni et al., 2021) under CC BY-NC-ND 4.0
- wav2vec 2.0 under Apache License 2.0
- LASER (Artetxe and Schwenk, 2019) under BSD
- Transformers (Wolf et al., 2020) under Apache License 2.0
- xnmt (Neubig et al., 2018) under Apache License 2.0
- Lightning (Falcon and The PyTorch Lightning team, 2019) under Apache License 2.0

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section "Limitations" after "Conclusion"

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract 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 3 and Appendix B.2

☑ B1. Did you cite the creators of artifacts you used?
Section 3 and Appendix B.2

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix B.2

☑ 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)?
Appendix B.2

□ 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?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3.1

☑ 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.
Section 3.1

C ☑ Did you run computational experiments?

Section 3 and Section 4

☑ 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 3.2 and Appendix B.1

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 3.2

☑ 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

☑ 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 3.2

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.