Inverted Projection for Robust Speech Translation

Dirk Padfield[*]
Google Research
padfield@google.com

Colin Cherry[*]
Google Research
colincherry@google.com

Abstract

Traditional translation systems trained on written documents perform well for text-based translation but not as well for speech-based applications. We aim to adapt translation models to speech by introducing actual lexical errors from ASR and segmentation errors from automatic punctuation into our translation training data. We introduce an inverted projection approach that projects automatically detected system segments onto human transcripts and then re-segments the gold translations to align with the projected human transcripts. We demonstrate that this overcomes the train-test mismatch present in other training approaches. The new projection approach achieves gains of over 1 BLEU point over a baseline that is exposed to the human transcripts and segmentations, and these gains hold for both IWSLT data and YouTube data.

1 Introduction

Speech translation is an important field that becomes more relevant with every improvement to its component technologies of automatic speech recognition (ASR) and machine translation (MT). It enables exciting applications like live machine interpretation (Cho and Esipova, 2016; Ma et al., 2019) and automatic foreign-language subtitling for video content (Karakanta et al., 2020).

However, translation of speech presents unique challenges compared to text translation. Traditional text translation systems are often trained with clean, well-structured text consisting of (source language, target language) sentence pairs gathered from text documents. This works well for translating written text, but for cascaded systems composed of speech → automatic transcription → automatic translation, errors from ASR and automatic punctuation are amplified as they pass through the translation system. Such systems suffer from three issues: 1) spoken language structure is different from written language structure and can include aspects like disfluencies and partial sentences, 2) ASR systems are not perfect and introduce errors in the stage from speech to source transcript, and 3) mistakes from automatic punctuation systems can lead to unnatural sentence segments and boundaries (Makhija et al., 2019; Nguyen et al., 2019; Wang et al., 2019). These problems can lead to poor translations and pose unique challenges for MT that are not readily addressed by current methods. In this work, we set out to make MT robust to the second and third issues in particular.

We have developed an approach to train translation models that are robust to transcription errors and punctuation errors, by introducing errors from actual ASR and automatic punctuation systems into the source side of our MT training data. This is similar in spirit to the method of Li et al. (2021), which introduces artificial sentence boundary errors into the training bitext. However, instead of artificial boundaries, our segmentation approach uses actual boundaries generated by the automatic punctuation system, which required the development of our inverted projection technique, and we also include errors from ASR. For a small subset of our training set, we assume access to long-form source audio documents, their corresponding human transcriptions, and translations of those transcriptions. This makes it possible to compare the performance of a baseline model trained on the human transcription with a model trained on source sentences derived from applying ASR transcription and automatic punctuation to the same audio.

Our primary contributions are first to show how to produce training data that captures the errors from automatic transcription and punctuation, which requires a non-trivial re-segmentation of the reference translation that we call *inverted projec-*

[*]equal contribution

Proceedings of the 18th International Conference on Spoken Language Translation, pages 236–244
Bangkok, Thailand (Online), August 5–6, 2021.

tion; and second to show experimentally that it is actually more important to expose the MT system to segmentation errors than lexical transcription errors when aiming for speech-robust MT.

2 Background

Compounding errors from ASR are known to cause problems when cascaded into MT (Ruiz et al., 2017). These issues are one of the main motivators for end-to-end modeling of speech translation (Weiss et al., 2017; Bansal et al., 2018; Sperber et al., 2019). However, we consider end-to-end modeling out of scope for this study since we aim to benefit from the modularity that comes with a cascaded speech translation strategy. To improve a cascade's robustness to speech input, one can train the MT system with some combination of artificial errors, actual ASR output, or long-form segmentation errors. We discuss each in turn.

Introducing artificial errors into the training set has the advantage of being efficient, and not necessarily tied to a specific ASR system. One can add Gaussian noise to the source embeddings (Cheng et al., 2018) or induce lexical substitutions that may be informed by phonetics (Li et al., 2018; Liu et al., 2019). Sperber et al. (2017) experiment with a noise model that can perform insertions, deletions and substitutions, but find little value in refining the substitutions to account for word frequency or orthographic similarity.

More related to our efforts are those that use actual ASR output. Early experiments used ASR output to replace the source side of parallel text during training (Post et al., 2013; Sperber et al., 2017). These did not perform well, likely because ASR word error rates (WER) on the Fisher Corpus were more than 40%, resulting in an unreliable training signal. Recently, Cheng et al. (2019) showed that, given ASR training corpora (coupled audio-transcription pairs), one can build a robust MT system by training with the normal MT objective on MT corpora, plus a mixture of: (1) an adversarial objective that tries to bring encoder representations for ASR output close to those of human transcriptions; and (2) a normal MT objective that has ASR output as source and machine translations of human transcripts as target. In an IWSLT TED translation scenario, they showed large improvements (+2.5 BLEU) using the second idea alone, which we take as a strong signal that there is much to be gained by training with ASR output on the source side.

Segment \ Token	Human	System
Human	Baseline	Token Robustness
System	Segment Robustness	System Robustness

Table 1: Combinations of segments and tokens.

We consider a long-form scenario where sentence boundaries for the input audio are not given at test time. As such, the method of Li et al. (2021) to make MT robust to segment boundary errors is very relevant. They introduce artificial sentence boundary errors in their training bitext. They first fragment adjacent source sentences, and then produce analogous fragments in the target according to proportional token lengths. We draw inspiration from their approach when building the target sides of our inverted projections.

3 Methods

Our approach to producing MT systems that are robust to automatic transcription errors is to introduce errors from our ASR system into our MT training data. Throughout the discussion of our methods, we make use of both human (manual) and system (automated) transcriptions of the source audio. When discussing the target-side of our training data, we use instead the term "gold" to indicate a trusted reference translation. Throughout our experiments, the gold standard is a human translation of the human transcript (Post et al., 2013; Sperber et al., 2017), though it could just as easily, and much less expensively, be a machine translation of the human transcript (Cheng et al., 2019).

We divide transcription errors into two categories: token and segment errors. A *token* error is any word that is transcribed incorrectly by ASR, such as a homophone substitution or the omission of a mumbled word. Meanwhile, *segment* errors are introduced by failing to correctly break the recognized text into sentence-like segments. A human transcription is expected to have error-free tokens and segments.

Table 1 presents a baseline and three ways to turn long-form Audio-Transcript-Translation triples into robust training data suitable for fine-tuning an NMT model. Training models with human tokens and segments is the common translation mode, so we mark it here as *Baseline*. Training

Human	I	checked	the	**weather**	–	this	evening	.	It	will	**rain**	tomorrow	.
System	I	checked	the	**whether**	.	This	evening	–	it	will	**rein**	tomorrow	.

Table 2: Our running example of human and system transcriptions, with the system having both lexical and segmentation errors. The Levenshtein alignment is given by column alignment, with – indicating insertion or deletion.

with system tokens and human segments is the approach taken by others such as (Cheng et al., 2019), resulting in *Token Robustness*. In the case of long-form ASR, the human segments can be projected onto the ASR output. This is an effective approach for exposing the training model to token errors from ASR, but it has an important disadvantage, as it results in a train-test mismatch because the human segments seen during training will not be available at inference time. We describe this approach in Section 3.2 to provide a comparison to our approaches using system segments and to introduce some of the concepts and tools used in those approaches.

The two approaches using system segments are the main innovations in this paper. Introducing segment errors alone results in *Segment Robustness* (Section 3.3), while segment and token errors together result in *System Robustness* (Section 3.4); that is, MT that is robust to the complete long-form transcription pipeline. We will show in the following sections how we can project system segments onto the source and target text; we call this an *inverted projection*.

3.1 Levenshtein Projection

A key component to all of the approaches in Table 1 is an alignment between the system (ASR) transcription and a human transcription of the same long-form audio. Inspired by common practice in evaluation for long-form speech translation (Matusov et al., 2005), we employ a token-level, case-insensitive Levenshtein alignment of the two transcripts. The Levenshtein alignment is monotonic, parameter-free, and its dynamic programming algorithm is fast enough to be easily applied to very long sequences. We show an example alignment in Table 2. By tracking the alignment of tokens immediately before segment boundaries (always end-of-sentence periods in our example), we can project segment boundaries from one transcription to another, which allows us to produce the various entries in Table 1, as we describe in more detail in the following subsections.

3.2 Token Robustness Training

The first approach to training on ASR sentences is straightforward and is a variant of a published result by Cheng et al. (2019). We Levenshtein-align the human transcript to the system transcript, and project the human sentence boundaries onto ASR. Since each human transcript is already paired with a gold standard translation, this projection makes it easy to align each projected ASR segment with a gold translation. We then train the model with (projected-ASR-source, gold translation) pairs. The Token Robustness training pair derived from our running example from Table 2 is shown in Table 3. The resulting source sentence, marked with $*$, has ASR token errors but human segment boundaries.

The main advantage of this approach is that it uses the gold translations as written; the model trains on well-formed translations. However, it suffers from a serious disadvantage: the model will only train on human segment boundaries, although at test time we will translate according to system segment boundaries, resulting in a train-test mismatch. Our experiments in Section 5 demonstrate that this is a serious drawback. In fact, when the WER is low, the token errors present in Token Robustness training are ignored by the model since they are overwhelmed by segment errors. In Section 3.3, we introduce an approach to overcome this limitation.

3.3 Segment Robustness Training

To address the segment-boundary train-test mismatch present in Token Robustness training, we can invert the projection and use system segments. That is, we project the system segment boundaries onto the human transcription.

System segments are derived from automatic punctuation and sentence splitting of the system transcription. As with Token Robustness, we Levenshtein-align the human transcript to the system transcript, but this time project the system segmentation onto the human transcript. Unlike the Token Robustness scenario, it is non-trivial to get

	Gold De	Ich	habe	heute	Abend	das	Wetter	überprüft	.	Morgen	wird	es	regnen	.
	Human En	i	checked	the	**weather**	this	evening	.		it	will	**rain**	tomorrow	.
$*$	System En	i	checked	the	**whether**	.	this	evening		it	will	**rein**	tomorrow	.

Table 3: Token Robustness ($*$). A Levenshtein alignment projects system tokens onto human segments. We have greyed out punctuation and lowercased to show the actual English text used in training.

	Gold De	Ich habe heute Abend *(I have this evening)*	das Wetter überprüft . Morgen wird es regnen . *(the weather checked . It will rain tomorrow .)*
$+$	Human En	i checked the **weather**	this evening . it will **rain** tomorrow .
$**$	System En	i checked the **whether** .	this evening it will **rein** tomorrow .

Table 4: Segment Robustness ($+$) and System Robustness ($**$). A Levenshtein alignment projects human tokens onto system segments, and then human-transcript-to-translation length ratios are used to align the German tokens to both. We have greyed out punctuation and lowercased to show the actual English text used in training.

corresponding segment boundaries for the gold-standard translations when training for Segment Robustness. We could perform a statistical word alignment between the human transcription and its translation to determine word-level interlingual semantic correspondence, but in similar situations such as prefix training for simultaneous translation (Niehues et al., 2018; Arivazhagan et al., 2020), this has not resulted in improvements over a simple proportional length-based heuristic. Therefore, we use human-transcript-to-translation length ratios (in tokens) to segment the gold translations so that their new segment lengths match the projected human source segment lengths. Finally, we train on (projected-human-source, projected-gold-translation) pairs. This is similar to how artificial target sentences were constructed by Li et al. (2021), but in our case, the boundaries are determined by automatic punctuation on ASR output, rather than from introducing boundary errors at random.

Table 4 shows the resulting human English and gold German segments for our running example; the source row marked with $+$ is used in Segment Robustness training. To illustrate the length-ratio token alignment, we can see that the total token length of the human English is 12, and the gold German is 13. The English is segmented into lengths 4 and 8, meaning the German is segmented to lengths $4/12 \cdot 13 = 4.33 \approx 4$ and $8/12 \cdot 13 = 8.66 \approx 9$. The resulting references will not always semantically match the content in the new source segments. In this example, they do not: an English gloss of the German shows that the semantics have diverged. But they are often close enough, and our hypothesis is that the benefit of exposure to realistic source fragments will outweigh the cost of occasional semantic misalignment. Furthermore, we use this robustness data only to fine-tune a system that has seen many semantically valid pairs.

3.4 System Robustness Training

In Section 3.3, the inverted projection approach was applied to the human transcripts. While this may seem unnatural, it provides a measure of the improvement that can be achieved by just adjusting the training set's source segment boundaries so that they match what the model will see during inference. Next, we build upon this approach by injecting the ASR token errors into the training data as well.

Training a model that sees both system token errors and segment boundary errors involves a slight variation on the setup in Section 3.3. We use the same alignment approach, but here we use it only to get projected gold translations since the system transcripts already have system segment boundaries. We then train the model with (system source, projected-gold-translation) pairs.

The main advantage of this approach is that the source side exactly matches the pipeline, completely bridging the train-test mismatch. The disadvantage, as in Section 3.3, is that the system segments may lead to fragmented or semantically misaligned reference sentences. Table 4 marks the source row used for System Robustness training with a $**$.

4 Experimental Setup

4.1 Data

We experiment on the IWSLT English to German (EnDe) speech translation scenario. We use the

IWSLT 2018 EnDe training data, including both the official training set and the leftover TED talks not included in any other test set, for a total of about 2400 talks and 0.25M sentence pairs. We found it beneficial to also include the 4.6M sentence pairs of the WMT 2018 EnDe corpus (Bojar et al., 2018) during training to increase our feasible MT model size and MT accuracy. For the IWSLT data, we scrape the ground truth transcripts and translations from www.ted.com directly because we found that the official IWSLT datasets omit transcriptions for many sentences. Since we are interested in long-form scenarios, we want to be sure to retain all sentences.

We evaluate our models on past IWSLT spoken language translation test sets. We use IWSLT tst2014 (Cettolo et al., 2014) as a dev set, which consists of 14 TED talks and about 1,200 sentences. We test on IWSLT tst2015 (Cettolo et al., 2015), which consists of 12 TED talks totalling about 1,200 sentences. Punctuated ASR transcriptions are obtained from the publicly available Speech-to-Text Google API[1]; using a separate ASR system in this way disconnects the ASR and NMT models, improving modularity. This achieves a WER of 5.5% on tst2015 ignoring case and punctuation. We run a sentence breaker on the punctuated source to determine the segments to be translated by NMT. Since these segments need not match the reference sentence boundaries, especially when punctuation is derived automatically on ASR output, we use our Levenshtein alignment as described in Section 3 to align our translation output with the gold-standard translation's segments before evaluating quality with case-sensitive BLEU (Matusov et al., 2005). All models are trained and tested on lowercased and unpunctuated versions of the source, as doing so is known to improve robustness to ASR output (Li et al., 2021).

4.2 Baseline

For all our experiments, we use a Transformer model (Vaswani et al., 2017) with a model dimension of 1024, hidden size of 8192, 16 heads for multihead attention, and 6 layers in the encoder and decoder. The models are regularized using a dropout of 0.3 and label smoothing of 0.1 (Szegedy et al., 2015). We use a shared SentencePiece tokenizer (Kudo and Richardson, 2018) with a 32k vocabulary. We decided on these settings through hyper-parameter tuning on the IWSLT dev set.

As a baseline, we train a model that includes a mix of WMT and human-transcribed IWSLT data, but with no ASR-transcribed IWSLT data. During training, for each batch, we sample 90% of data from WMT and 10% from IWSLT. This mixture was chosen based on the best performance of a grid-search of weightings between these two datasets evaluated on the IWSLT dev set. Because this baseline has already seen the human transcripts and translations of the IWSLT data, it has already adapted its domain to both news and TED data. By ensuring that this baseline has already adapted, we are able to isolate the effects of ASR errors and segmentation errors on the fine-tuned models. We train the model using pairs of (source, target) sentences, where target German translations are untouched, retaining case and punctuation.

4.3 Model fine-tuning

Starting from the baseline, we fine-tune the model on data from each scenario, each time starting from the same checkpoint of the baseline. The best-performing checkpoint of each fine-tuning experiment is chosen based on the BLEU score on the dev set, and this checkpoint is used to evaluate on the test set. Fine-tuning is about 35x faster than training from scratch in our configuration and converges after running through less than 5 epochs of the IWSLT data (≈0.25M sentence pairs). We repeat each experiment multiple times to account for any variations in the runs.

4.4 Filtering

All of the processing steps described so far have included all of the ASR sentences, regardless of their quality. However, some ASR sentences have high WER compared with the human transcripts. This happens when, for example, the ASR transcribes a video playing in the background that was not included in the gold transcript. These examples can be so egregious that they can confuse the model. To filter the dataset, we remove only from our *training set* all ASR sentences with WER $\geq 50\%$ as compared with the human transcripts; this removes approximately 4% of the training data. The sentences with WER between 0% and 50% are useful because they demonstrate ASR errors relative to human transcripts but not egregious errors. We include results on this filtered set as an additional row in our results tables. Note that the filtering is only applied to the training data and is not applied on

[1] http://cloud.google.com/speech-to-text

the test set since we wouldn't have access to WER during inference time. This should not be confused with the average WER measured on each test set, which is 5.5% for IWSLT (see Table 5) and 9.0% for YouTube (see Table 6), which is an indicator of the quality of the NMT model's input source sentences generated by the ASR system.

5 Results

5.1 IWSLT results

Table 5 compares the results of the different combinations of segments and tokens from Table 1. For the test set, automatic punctuation is first applied and used to split the ASR output into sentences, and then it is stripped of case and punctuation. Sentences are translated one at a time with whatever system is under test. The checkpoint is chosen according to the dev set for each scenario, and the resulting BLEU scores on the test set are presented in the "ASR" column. For completeness, we also compute the BLEU score on the IWSLT human transcripts using the same model and checkpoint and report it in the "HT" column. As expected, this "HT" score decreases with increasing adaptation to the system tokens and segments, but this does not affect our results because, during inference, our system will only be applied to ASR sentences with automatic punctuation.

The baseline, trained from scratch using the human tokens and human segments (WMT + IWSLT), achieves a score of 26.5 BLEU points on the ASR set. As described in Section 4.2, this baseline training uses only 10% IWSLT data. Since the fine-tuning experiments use 100% IWSLT data, those models are arguably more adapted to the TED domain, which could contribute to any improvements over the baseline. To control for this, we fine-tuned an additional model on 100% human token, human segment IWSLT data, but this yielded no improvement over the baseline, likely because the baseline has already seen this IWSLT data during training. Thus, we didn't include this experiment in Table 5.

All of the fine-tuning experiments in Table 5 start with the baseline from the first row, which was trained without knowledge of the ASR transcripts. The Token Robustness experiment starts from the baseline and fine-tunes on ASR; it shows no improvement compared to the baseline, which indicates that the ASR errors are sufficiently subtle compared to the segment errors so that the model cannot adapt to them. On the other hand, the last 3 rows demonstrate significant gains when the text is projected using the system segments. In particular, the System Robustness experiment shows an improvement over the Segment Robustness, and the best results are achieved with System Robustness when removing ASR transcripts with high WER. This yields a gain of more than 1 BLEU point over the baseline. This indicates that, once the train-test segment mismatch has been corrected for, the model is able to adapt to and correct the subtle ASR errors. These improvements indicate the value of making the segmentation errors visible to NMT training using the two steps of projecting source and re-aligning translation.

Training condition	HT	ASR
Baseline (human tokens and segments)	33.6	26.5
Token Robustness (ASR source, human segments)	32.7	26.0
Segment Robustness (human source, system segments)	32.1	27.1
System Robustness (ASR source, system segments)	32.1	27.4
System Robustness (ASR source with WER $\leq$ 50%, system segments)	32.3	27.6

Table 5: Results on IWSLT tst2015 data. HT stands for "human transcript". All numbers represent the translation BLEU, and each score is the average across 3 runs. The ASR WER on the test sentences is 5.5%.

The fact that our Token Robustness model does not improve over the baseline is likely because there are very few lexical errors since our ASR model for English is very good, with a mean WER of 5.5%. This is true even when we use the approach from Section 4.4 to remove high WER ASR sentences during training (results not included in Table 5). This is in contrast to the results of Cheng et al. (2019), which demonstrated improvements using ASR with human segments. Those results, however, were achieved with the ASR model provided by IWSLT 2018, which has a much worse WER than the ASR used in our work.[2] We likely could have replicated their result had we used a weaker ASR model.

Our Segment Robustness approach and dataset are similar to the synthetic segment breaks ap-

[2] Zenkel et al. (2018) report that the IWSLT 2018 ASR has a WER of 22.8% on IWSLT tst2014, while the ASR used in our experiments achieves a WER of 8.0% on the same set.

Training condition	HT	ASR
Baseline (human tokens and segments)	30.3	25.4
Token Robustness (ASR source, human segments)	29.8	25.1
Segment Robustness (human source, system segments)	29.3	26.6
System Robustness (ASR source, system segments)	29.3	26.4
System Robustness (ASR source with WER $\leq$ 50%, system segments)	29.4	26.6

Table 6: Results on 88 English videos from YouTube translated into German. No new models were trained in these experiments: the models trained in Table 5 were directly evaluated on these videos. The ASR WER on the test sentences is 9.0%.

proach in (Li et al., 2021). According to Table 5, our results yielded a BLEU score of 27.1, which is similar to the score of 27.0 reported in Table 4 of that paper, which represents their best result from training with synthetic segment breaks.

5.2 YouTube results

To test the generalization of our approach, we applied the models trained on the IWSLT data in Section 5.1 to another dataset consisting of 88 English videos selected from YouTube. The videos are selected to have a single speaker, and are truncated to a length of roughly 1 minute, perhaps interrupting a sentence. Each of the 920 sentences in the human transcription of these videos was professionally translated into German.

No new models were trained in this section; every line in Table 6 is a corresponding system from Table 5. For each of the experiments, we take the corresponding model trained on IWSLT and test it on this new YouTube EnDe test set. This enables us to determine the generalization ability of the approach.

According to Table 6, the model performs remarkably similar on this YouTube dataset. In particular, the improvement over the baseline of the System Robustness in the last row is about 1.2 BLEU points, comparable to the 1.1 BLEU point improvement in Table 5.

Note that, because the models were fine-tuned on the IWSLT ASR dataset starting from a mix of WMT and IWSLT, there is a domain mismatch between this training data and the YouTube testing data. Nevertheless, the System Robustness approach shows a clear improvement. Thus, we expect that if we trained a model directly on YouTube data, we would see even higher BLEU scores. This is a task for future work.

6 Conclusions

To aid text-based translation models to adapt to speech data, we introduced an inverted projection approach that projects automatically detected system segments onto human transcripts and then re-segments the gold translations to align with the projected human transcripts. This approach overcomes the train-test mismatch present in previous attempts to train on long-form ASR output by exposing MT training to both token and segment errors, exactly matching the source transcription pipeline used at test time. The results demonstrate a gain of over 1 BLEU point on both IWSLT data and YouTube data.

For future work, we aim to train models on languages with higher ASR WER since our English WER is very low (5.5%). We also plan to experiment with MT targets during training to address the data bottleneck. And we also plan to investigate whether we can eliminate segmentation altogether with document-level speech translation.

Acknowledgments

We would like to thank Naveen Ari, Te I, and Wolfgang Macherey for their deep discussions about this work and for their helpful suggestions while reviewing our paper.

References

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Low-resource speech-to-text translation. In *Proceedings of INTERSPEECH*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Bel-

gium, Brussels. Association for Computational Linguistics.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The IWSLT 2015 evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT)*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT)*.

Qiao Cheng, Meiyuan Fang, Yaqian Han, Jin Huang, and Yitao Duan. 2019. Breaking the data barrier: Towards robust speech translation via adversarial stability training. In *International Workshop on Spoken Language Translation (IWSLT)*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *CoRR*, abs/1606.02012.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020. Is 42 the answer to everything in subtitling-oriented speech translation? In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 209–219, 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.

Daniel Li, Te I, Naveen Arivazhagan, Colin Cherry, and Dirk Padfield. 2021. Sentence boundary augmentation for neural machine translation robustness. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7553–7557.

Xiang Li, Haiyang Xue, Wei Chen, Yang Liu, Yang Feng, and Qun Liu. 2018. Improving the robustness of speech translation. *CoRR*, abs/1811.00728.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

K. Makhija, T. Ho, and E. Chng. 2019. Transfer learning for punctuation prediction. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 268–273.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *International Workshop on Spoken Language Translation (IWSLT)*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. In *Proceedings of INTERSPEECH*, pages 1293–1297.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome spanish–english speech translation corpus. In *International Workshop on Spoken Language Translation (IWSLT)*.

Nicholas Ruiz, Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. 2017. Assessing the tolerance of neural machine translation systems against speech recognition errors. In *Proc. Interspeech 2017*, pages 2635–2639.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2019. Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*, 7:313–325.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2017. Toward robust neural machine translation for noisy input sequences. In *International Workshop on Spoken Language Translation (IWSLT)*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. 2019. Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 1–11, Dublin, Ireland. European Association for Machine Translation.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly transcribe foreign speech. In *Proceedings of INTERSPEECH*.

Thomas Zenkel, Matthias Sperber, Jan Niehues, Markus Müller, Ngoc-Quan Pham, Sebastian Stüker, and Alex Waibel. 2018. Open source toolkit for speech to text translation. *Prague Bull. Math. Linguistics*, 111:125–135.