Does Joint Training Really Help Cascaded Speech Translation?

Viet Anh Khoa Tran David Thulke Yingbo Gao Christian Herold Hermann Ney
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{vtran|thulke|gao|herold|ney}@i6.informatik.rwth-aachen.de

Abstract

Currently, in speech translation, the straightforward approach - cascading a recognition system with a translation system - delivers state-of-the-art results. However, fundamental challenges such as error propagation from the automatic speech recognition system still remain. To mitigate these problems, recently, people turn their attention to direct data and propose various joint training methods. In this work, we seek to answer the question of whether joint training really helps cascaded speech translation. We review recent papers on the topic and also investigate a joint training criterion by marginalizing the transcription posterior probabilities. Our findings show that a strong cascaded baseline can diminish any improvements obtained using joint training, and we suggest alternatives to joint training. We hope this work can serve as a refresher of the current speech translation landscape, and motivate research in finding more efficient and creative ways to utilize the direct data for speech translation.

1 Introduction

Speech translation (ST) is the task of automatic translation of speech in some source language into some other target language (Stentiford and Steer, 1988; Waibel et al., 1991). Traditionally, a cascaded approach is used, where an automatic speech recognition (ASR) system is used to transcribe the speech, followed by a machine translation (MT) system, to translate the transcripts (Sperber and Paulik, 2020). The problem of error propagation has been the center of discussion in ST literature (Ney, 1999; Casacuberta et al., 2004; Matusov et al., 2005; Peitz et al., 2012; Sperber et al., 2017), and instead of using the discrete symbols in the source languages, ideas like using n-best lists, lattices, and neural network hidden representations are investigated (Saleem et al., 2004; Kano et al., 2017; Anastasopoulos and Chiang, 2018; Zhang et al., 2019; Sperber et al., 2019). For a more systematic review of the ST development, we refer the readers to Sperber and Paulik (2020).

With recent efforts in the expansion of the ST data collection (Di Gangi et al., 2019; Beilharz et al., 2020), more and more direct ST data is available. Such direct data comes as pairs of source speech and target translation, and often as triplets further including source transcriptions.

Various joint training methods are proposed to use such data to improve cascaded systems, with the hope that uncertainties during transcription can be passed on to translation to be resolved. Here, what we call "joint training" is often referred to as "end-to-end training" in the literature, where the direct ST data is utilized in the joint optimization of the ASR and MT models (Kano et al., 2017; Berard et al., 2018; Anastasopoulos and Chiang, 2018; Inaguma et al., 2019; Sperber et al., 2019; Bahar et al., 2019; Wang et al., 2020; Bahar et al., 2021). In this work, we revisit the principal question of whether or not joint training really helps cascaded speech translation.

2 Cascaded Approach

In traditional cascaded systems, an ASR model $p(f_1^J|x_1^T)$ and an MT model $p(e_1^I|f_1^J)$ are trained separately, where we denote speech features as x_1^T, transcriptions as f_1^J, and translations as e_1^I. The decoding is done in two steps:

$$\hat{f}_1^J = \underset{[f_1^J]}{\operatorname{argmax}}\, p(f_1^J|x_1^T)$$
$$\hat{e}_1^I = \underset{[e_1^I]}{\operatorname{argmax}}\, p(e_1^I \mid \hat{f}_1^J)$$

The argmax is approximated using beam search for computational reasons, and we will assume a fixed beam size N for the decoding of both transcriptions and translations.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4480–4487

3 Joint Training Approaches

3.1 Top-K Cascaded Translation

Assume we have pre-trained an ASR and an MT model, and some direct ST training data is available. The pre-trained ASR model is used to produce a K-best list of ASR hypotheses $F_1, F_2, \ldots, F_K$ using beam search with beam size $N \geq K$. While there is no unique method to make use of the top-K transcript, we describe **Top-K-Train**, a straightforward algorithm similar to re-ranking. We obtain the score $\tilde{p}$ for each ASR hypothesis with length normalization and normalize them locally within the top-K hypotheses.

$$p(F_k|x_1^T) = \frac{\tilde{p}(F_k|x_1^T)}{\sum_{k'=1}^{K} \tilde{p}(F_{k'}|x_1^T)} \quad (1)$$

During training, $p(e_i|F_K; e_0^{i-1})$ is the MT model output. Given the ASR hypotheses $F_1, \ldots, F_K$, the following training objective is maximized.

$$\log \left(\sum_{k=1}^{K} p(F_k|x_1^T) \prod_{i=1}^{I} p(e_i|F_k; e_0^{i-1}) \right)$$

We hypothesize that this objective (a) exposes different transcriptions and potential ASR errors to the MT model and (b) encourages the ASR model to produce hypotheses closer to the expectations of the MT model, thus reducing model discrepancy. Since discrete ASR hypotheses are passed to the MT model from a previous beam search, the error signal to ASR is passed via the renormalized transcript scores during backpropagation.

Similarly, we introduce **Top-K-Search**. We obtain an MT hypothesis E_k for each F_k using beam search. The final hypothesis is obtained as below.

$$\hat{e}_1^{\hat{I}} = \underset{E_k}{\operatorname{argmax}} \{p(F_k|x_1^T) \cdot p(E_k|F_k)\}$$

Here, $p(F_k|x_1^T)$ is obtained as in Equation 1 and $p(E_k|F_k)$ is the length normalized translation score from the MT model. Observe that this search is applicable to any cascade architecture and is thus independent of the training criterion. In our experiments, we always Top-K-Search when decoding models trained with Top-K-Train. The idea of generating the top-K ASR hypotheses during search has also been explored in the literature (e.g. Section 3.3).

3.2 Tight-Integration

Another way to train the cascade architecture using direct ST data is the **tight integrated cascade** approach (Bahar et al., 2021). We introduce an exponent γ that controls the sharpness of the distribution of the conditional probabilities. Thus, instead of passing the 1-best hypothesis of the ASR system as a sequence of 1-hot vectors, we pass the renormalized probabilities to the MT model.

$$p(f_j|f_1^{j-1}; x_1^T) = \frac{\tilde{p}^{\gamma}(f_j|f_1^{j-1}x_1^T)}{\sum_{f' \in |V_F|} \tilde{p}^{\gamma}(f'_j|f_1^{j-1}x_1^T)}$$

Here, V_F is the vocabulary of the ASR system.

3.3 Searchable Hidden Intermediates

Dalmia et al. (2021) propose passing the final decoder representations of the N-best ASR hypotheses (i.e. the **searchable hidden intermediates**) directly to the MT system, bypassing the MT input embedding.

Additionally, they extend the multi-task learning approach by allowing the MT decoder to attend to the ASR encoder states, which in turn are optimized using beam search in training. They show that during decoding, a higher ASR beam size indeed leads to a better ST performance.

4 Experimental Results and Analysis

We focus on the MuST-C English-German speech translation task (Di Gangi et al., 2019) in the domain of TED talks and evaluate on test-HE and test-COMMON. We use an in-house filtered subset of the IWSLT 2021 English-German dataset as in Bahar et al. (2021), which contains 1.9M segments (2300 hours) of ASR data and 24M parallel sentences of MT data. The in-domain ASR data comprises MUST-C, TED-LIUM, and IWSLT TED, while the out-of-domain ASR data consists of EuroParl, How2, LibriSpeech, and Mozilla Common Voice. For translation, the dataset contains 24M parallel sentences of in-domain translation data (MuST-C, TED-LIUM, and IWSLT TED), as well as out-of-domain translation data (NewsCommentary, EuroParl, WikiTitles, ParaCrawl, CommonCrawl, Rapid, OpenSubtitles2018). For ST data, we only use MuST-C. We provide further details in Appendix A. Depending on whether or not fine-tuned on in-domain ASR and MT data, we split our experiments into two sets: A1-A5 and B1-B5.

	Model	Data			MuST-C En-De			
		MT	ASR	ST	tst-HE		tst-COMMON	
					BLEU	TER	BLEU	TER
	Bahar et al. (2021)							
	cascade	✓	✓	-	25.0	59.2	25.9	56.2
	tight integrated cascade	✓	✓	✓	26.8	57.2	26.5	54.8
	Xu et al. (2021)							
	cascade	-	-	✓	-	-	23.3	-
	cascade	✓	✓	✓	-	-	28.1	-
	direct + pretraining + multi-task	-	-	✓	-	-	25.2	-
	direct + pretraining + multi-task	✓	✓	✓	-	-	28.1	-
	Dalmia et al. (2021)							
	searchable hidden intermediates	-	-	✓	-	-	26.4	-
	Inaguma et al. (2021)							
	cascade	✓	✓	-	26.1	-	29.4	-
	direct + KD	✓	✓	✓	27.4	-	30.9	-
	searchable hidden intermediates + KD	✓	✓	✓	-	-	30.8	-
	this work (no fine-tuning)							
A1	ground-truth transcript MT	✓	(✓)	-	30.4	54.0	32.5	48.9
A2	cascade	✓	✓	-	27.7	57.8	29.0	53.6
A3	tight integrated cascade	✓	✓	✓	28.7	56.1	29.2	53.5
A4	Top-K-Train	✓	✓	✓	28.7	56.9	30.0	52.5
A5	Top-K-Search	✓	✓		28.2	57.1	29.4	53.1
	this work (fine-tuned ASR and MT)							
B1	ground-truth transcript MT	✓	(✓)	-	31.1	53.2	33.7	48.2
B2	cascade	✓	✓	-	29.1	56.6	30.5	52.2
B3	tight integrated cascade	✓	✓	✓	29.4	55.9	30.1	52.5
B4	Top-K-Train	✓	✓	✓	29.4	55.9	30.5	51.9
B5	Top-K-Search	✓	✓	-	29.4	56.4	30.8	51.9

Table 1: Results measured in BLEU [%] and TER [%] on the MuST-C En-De task. Fine-tuning refers to additional phase of training exclusively on the in-domain subset of the training data of both ASR and MT models.

Without fine-tuning, we observe that both the tight integrated cascade (Bahar et al., 2021) (A3) and our marginalization approach (A4) outperform the baseline cascade (A2) on both test-HE and test-COMMON (Table 1). However, after fine-tuning both ASR and MT models, we do not observe significant improvements of the joint training methods (B3, B4) over the cascade baseline (B2) anymore.

Our experimental results suggest that the joint training of cascaded speech translation models does not seem to be effective. This poses the questions: why is that and what were we trying to achieve with joint training anyways? Sperber and Paulik (2020) highlighted three main issues with ST, and in the following, we will discuss these issues from the perspective of joint training.

Mismatched spoken and written domains Transcripts and translation data usually differ in e.g. linguistic style and punctuation. This mismatch poses a challenge for cascaded models, as translation models may struggle to handle transcript-style text. As Sperber and Paulik (2020) point out, some of the issues such as differences in punctuation can already be tackled by plain text normalization.

More generally, one can fine-tune the models on in-domain transcript-like ASR, MT, and ST data. It is unusual to find ST datasets that do not come with corresponding ASR and MT data, as ST data acquisition usually involves translating from transcriptions. Thus, we can simply fine-tune the ASR and MT models on these in-domain datasets.

Our results suggest that fine-tuning the ASR and MT models is comparable or even superior to fine-tuning these models in an end-to-end fashion on the respective speech translation dataset. However, Inaguma et al. (2021) and Bahar et al. (2021) report

significant improvements of their joint cascaded approach, which is similar to the tight integrated cascade, over their cascaded baseline (Table 1).

What is the reason for this disparity? We pin it down to one major difference: the use of in-domain ST data, or more precisely, the lack thereof. Inaguma et al. (2021) report that by fine-tuning on MuST-C and ST-TED, they are unable to significantly improve their MT baseline, and thus, the MT component in their cascade model is not fine-tuned on the domain of TED talks. In contrast, we find that in-domain fine-tuning significantly improves our cascaded model, especially if applied to both the ASR and MT models (Table 2), also improving the individual components, as we observe a decrease in WER [%] from 9.3 to 8.1 of the ASR component and an increase in BLEU [%] from 32.5 to 33.7 of the MT component on tst-COMMON. As pointed out earlier, fine-tuning diminishes any improvement obtained using any of the joint training methods we implement, as the cascade baseline significantly gains in performance.

Thus, in-domain fine-tuning is essential in order to tackle the disparity between the spoken and the written domain for vanilla cascaded models. This especially holds true for the MT model, which is trained on non-transcript-like data, but we want it to adapt to transcript-like inputs and transcript-like outputs (with punctuations, casing, etc.).

Intuitively, instead of in-domain fine-tuning, a simple remedy is to only use in-domain data for ASR and MT. Xu et al. (2021) observe an improvement of their multi-task learning method when allowing only MuST-C data (Table 1). However, the improvement vanishes as they introduce external ASR and MT data. Since the latter represents a more realistic data scenario, we believe that the inclusion of external ASR and MT data is necessary to obtain meaningful results.

In our fine-tuning setup, we do not only adapt to transcript-like data, but also to the specific domain of TED talks. In the literature, ST performance is commonly evaluated on test sets in the same domain as the ST data, e.g. TED talks (i.e. MuST-C or IWSLT TED), LibriSpeech (Kocabiyikoglu et al., 2018) and CallHome (Post et al., 2013). However, this poorly reflects real-life data conditions, because large amounts of in-domain ST data are not always available, while in-domain ASR and MT data is more accessible. As a consequence, we believe that end-to-end models are artificially favored over cascade models in these setups. We thus motivate future research to also consider out-of-domain or general-domain ST datasets, while allowing in-domain ASR and MT data.

	Fine-tuning		MuST-C En-De	
	ASR	MT	tst-HE	tst-COMMON
Inaguma et al. (2021)	-	-	26.1	29.4
Our work	-	-	27.7	29.0
	✓	-	27.9	29.5
	✓	✓	29.1	30.5

Table 2: Performance of our cascaded system under different in-domain fine-tuning conditions, results measured in BLEU [%].

Error propagation In case the ASR produces an error in the transcript or intermediate representations, this error is propagated to the translation model, which does not have any knowledge about the transcription process. Sperber and Paulik (2020) discuss this phenomenon under the term *erroneous early decisions*.

Intuitively, a remedy for this issue is joint training, as we allow the MT component to learn to use information that is missing or erroneous in the intermediate representation. For example, the tight integrated approach (Bahar et al., 2021) addresses this issue by expressing uncertainties in the transcription as posterior probabilities, while Dalmia et al. (2021) propose speech attention, i.e. a Transformer cross-attention sub-module in the MT component, attending over ASR encoder representations.

However, we make the case that joint training is not necessarily the only remedy to error propagation. Therefore, we consider a cheating experiment based on Top-K-Search. For each of the top-4 ASR hypotheses, we pick the translations generated by the MT model based on the sentence-level BLEU with the ground-truth target. In these experiments, we obtain a BLEU [%] score of 32.4 on tst-HE and 34.2 on tst-COMMON, in both cases outperforming the oracle MT using ground-truth transcripts. Thus, on average, the translation of one of the top-4 transcripts generated by the ASR model is no worse than using the ground-truth transcript (B1). Hence, we posit that error propagation can be alleviated by plain ASR re-ranking. A possible starting point is our Top-K-Search (A5, B5), giving small, but consistent improvements over their respective cascade baselines (A2, B2) without any joint training.

Similarly, instead of sequence-level reranking using the joint ASR-MT score, Dalmia et al. (2021)

propose augmenting the token-level ASR probabilities with either the CTC scores from the ASR encoder or an external LM score during beam search to incorporate ASR uncertainties.

Information loss Information loss occurs as the ASR model does not pass on auditory information such as intonation and timing, which may be relevant for the translation component. While we have no empirical evidence on the significance of such information on the final performance, we observe that our MT model, given the ground-truth transcripts, still significantly outperforms any speech translation model we investigate, doing so without any auditory information. Furthermore, our cheating experiments suggest one may even improve over ground-truth transcriptions without any auditory information. We posit that as of now, focusing on closing that gap is of higher importance.

5 Conclusion

In this work, we analyzed several reasons why joint training does not appear to help when individual automatic speech recognition and machine translation components are stronger. We point out that cascaded models achieve state-of-the-art performance when fine-tuned on in-domain ST data.

While we do not suggest that joint training is not worth exploring, we want to encourage future research to consider data conditions more carefully and produce strong cascade baselines. Concretely, we suggest (1) the inclusion of external ASR and MT data, (2) training strong cascade baselines by fine-tuning both ASR and MT models on in-domain transcript-like data, if available and (3) the investigation of data conditions where only out-of-domain ST data is available, while allowing in-domain ASR and MT data.

Limitations

In our experiments, we focus only on the MuST-C En-De task due to computational constraints. Further experiments on different language pairs could e.g. show differences of how spoken-written domain mismatch affects different languages. Again, experiments in different domains without direct ST data could further underline or refute our conclusions.

In our analysis, we include experimental results from other authors as we do not have the resources to reproduce every method. It is possible that differences in data filtering, data preprocessing, architectural choices, etc. could affect the comparability of these results.

We have only analyzed a subset of the joint cascade methods described in literature. A systematic overview of such methods is outside the scope of our work.

In order to be comparable to other works in literature, we mostly draw our conclusions using BLEU and TER. We acknowledge that using other automatic evaluation metrics and making use of human evaluation would strengthen the significance of our findings.

Acknowledgements

This work was partially supported by the project HYKIST funded by the German Federal Ministry of Health on the basis of a decision of the German Federal Parliament (Bundestag) under funding ID ZMVI1-2520DAT04A, and by NeuroSys which, as part of the initiative "Clusters4Future", is funded by the Federal Ministry of Education and Research BMBF (03ZU1106DA).

We thank Parnia Bahar for sharing her data preprocessing and training recipes.

References

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, New Orleans, Louisiana. Association for Computational Linguistics.

Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. 2021. Tight Integrated End-to-End Training for Cascaded Speech Translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 950–957.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. On using SpecAugment for end-to-end speech translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Benjamin Beilharz, Xin Sun, Sariya Karimova, and Stefan Riezler. 2020. LibriVoxDeEn: A corpus for German-to-English speech translation and German speech recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3590–3594, Marseille, France. European Language Resources Association.

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

Francisco Casacuberta, Hermann Ney, Franz Josef Och, Enrique Vidal, Juan Miguel Vilar, Sergio Barrachina, Ismael Garcıa-Varea, David Llorens, César Martınez, Sirko Molau, et al. 2004. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech & Language*, 18(1):25–47.

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. 2021. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1882–1896.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2012–2017. Association for Computational Linguistics.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 570–577.

Hirofumi Inaguma, Brian Yan, Siddharth Dalmia, Pengcheng Guo, Jiatong Shi, Kevin Duh, and Shinji Watanabe. 2021. Espnet-st iwslt 2021 offline speech translation system. *IWSLT 2021*, page 100.

Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. 2017. Structured-based curriculum learning for end-to-end english-japanese speech translation. In *INTERSPEECH*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE.

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*.

Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. 2018. Augmenting librispeech with French translations: A multimodal corpus for direct speech translation evaluation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Evgeny Matusov, Hermann Ney, and Ralf Schlüter. 2005. Phrase-based translation of speech recognizer word lattices using loglinear model combination. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 110–115. IEEE.

Hermann Ney. 1999. Speech translation: Coupling of recognition and translation. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 517–520. IEEE.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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.

Stephan Peitz, Simon Wiesler, Markus Nußbaum-Thom, and Hermann Ney. 2012. Spoken language translation using automatically transcribed text in training. In *Proceedings of the 9th International Workshop on Spoken Language Translation: Papers*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

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 *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany.

Shirin Saleem, Szu-Chen Jou, Stephan Vogel, and Tanja Schultz. 2004. Using word lattice information for a tighter coupling in speech translation systems. In *Proc. Int. Conf. on Spoken Language Processing*, pages 41–44.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics (ACL).

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2017. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1380–1389, Copenhagen, Denmark. Association for Computational Linguistics.

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 and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Fred WM Stentiford and Martin G Steer. 1988. Machine translation of speech. *British Telecom technology journal*, 6(2):116–122.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Waibel, Ajay N Jain, Arthur E McNair, Hiroaki Saito, Alexander G Hauptmann, and Joe Tebelskis. 1991. Janus: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 793–796. IEEE Computer Society.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020. Bridging the gap between pretraining and fine-tuning for end-to-end speech translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9161–9168.

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 *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 2619–2630.

Pei Zhang, Niyu Ge, Boxing Chen, and Kai Fan. 2019. Lattice transformer for speech translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6475–6484, Florence, Italy. Association for Computational Linguistics.

A Experimental Setup

For the ASR data, we extract 80-dimensional MFCC features every 10ms. The text data is post-processed by lower-casing, removing punctuation and transcriber tags, and by applying BPE (Sennrich et al., 2016) e a vocabulary size of 8000. We post-process the source text to be transcript-like, by removing punctuation and lower-casing. On both source and target sentences, we apply BPE with a vocabulary size of 32k. For models using tight integration, a separate translation model is trained using the transcription vocabulary. In both cases, the validation set is the concatenation of the validation sets provided by MuST-C and IWSLT TED.

Our implementation is based on fairseq (Ott et al., 2019) and is available online[1].

ASR model For speech recognition, we use a Transformer model (Vaswani et al., 2017) with 12 encoder layers and 6 decoder layers. Instead of absolute positional encodings, we use relative positional encodings as introduced by (Shaw et al., 2018). We reduce the audio feature length using a two-layer convolutional network with kernel size 5 and 1024 channels. Other parameters are adapted from the original base configuration. Our ASR model consists of 70M trained parameters.

Each epoch is split into 20 sub-epochs. We use the Adam optimizer (Kingma and Ba, 2015) with learning rate 0.0003 and 10 warmup sub-epochs, and the learning rate is scaled by 0.8 for every 3 sub-epochs without improvement on the validation set. As regularization, we use SpecAugment (Park et al., 2019) ($(F, m_F, T, m_T, p, W) = (16, 4, 40, 2, 1.0, 0)$), a dropout (Srivastava et al., 2014) of 0.1, and label smoothing (Pereyra et al., 2017) with $\alpha = 0.1$. Additionally, we train a CTC loss as an additional task during training (Kim et al., 2017).

MT model For translation, we also use a Transformer model with 6 encoder and 6 decoder layers. Again, we use relative positional encodings, other parameters are adapted from the original base configuration, resulting in a model with 70M trained parameters.

We use the same optimization procedure as with the ASR model, except that we now start with 4000

[1] https://github.com/tran-khoa/joint-training-cascaded-st

warmup steps. As regularization, we use a dropout of 0.1 and label smoothing with $\alpha = 0.1$.

Joint Training We use the same training setup as for the ASR model, but with a learning rate of 3×10^{-6}. Furthermore, as we are now working on a smaller ST dataset, an epoch is split into 10 sub-epochs.

Fine-tuning We fine-tune the ASR model on all in-domain ASR data (600K segments) with a learning rate of 0.0001, while the MT model is fine-tuned on the MuST-C dataset (300K parallel sentences) with a learning rate of 0.00001.

We use beam search with $N = 12$. All BLEU scores reported are calculated on case-sensitive data using the official WMT scoring script. TER scores are calculated using TERCom. For top-K experiments, we use $K = 4$.

We train our experiments on NVIDIA GTX 1080 Ti. Training the ASR takes around 4 weeks, all other experiment take around 2-3 weeks. The average runtime for inference on non-joint experiments is about 15 minutes, where joint experiments need around 2 hours.