INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

ASR Posterior-based Loss for Multi-task End-to-end Speech Translation

Yuka Ko[1], *Katsuhito Sudoh*[1,2], *Sakriani Sakti*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN Center for Advanced Intelligence Project (AIP), Japan
{ko.yuka.kp2, sudoh, ssakti, s-nakamura}@is.naist.jp

Abstract

End-to-end speech translation (ST) translates source language speech directly into target language without an intermediate automatic speech recognition (ASR) output, as in a cascading approach. End-to-end ST has the advantage of avoiding error propagation from the intermediate ASR results, but its performance still lags behind the cascading approach. A recent effort to increase performance is multi-task learning using an auxiliary task of ASR. However, previous multi-task learning for end-to-end ST using cross entropy (CE) loss in ASR-task targets one-hot references and does not consider ASR confusion. In this study, we propose a novel end-to-end ST training method using ASR loss against ASR posterior distributions given by a pre-trained model, which we call ASR posterior-based loss. The proposed method is expected to consider possible ASR confusion due to competing hypotheses with similar pronunciations. The proposed method demonstrated better BLEU results in our Fisher Spanish-to-English translation experiments than the baseline with standard CE loss with label smoothing.

Index Terms: end-to-end speech translation, multi-task learning, spoken language translation

1. Introduction

Speech translation (ST) systems translate source language speech to target language text. A simple approach is to cascade automatic speech recognition (ASR) and machine translation (MT), but this propagates ASR errors to MT. It is crucial to develop a robust MT system against ASR errors. A major approach to tackling this problem is to consider many ASR hypotheses in N-best and lattice form [1].

Recent ST studies attempt end-to-end ST without an explicit ASR module to obtain source language speech transcripts. The end-to-end approach is promising because it is basically free from ASR error propagation. However, the translation performance of a simple end-to-end ST trained using source language speech and target language translations is usually worse than a cascade ST. Multi-task learning [2, 3, 4, 5] is a promising approach to filling the gap between cascade and end-to-end ST by the use of an additional ASR-based loss function during training as an ASR subtask. Loss function for the ASR-task is usually implemented using cross entropy (CE) loss against reference transcriptions. However, the previous multi-task learning for end-to-end ST does not consider ASR confusion in ASR-task training. Since our objective is to obtain correct translations, we do not always have to make such hard decisions in an auxiliary ASR-task.

In this study, we propose a novel training method for end-to-end ST using ASR loss against given by a pre-trained ASR model as reference for prediction in an ASR subtask. We call this loss function *ASR posterior-based loss*. We can include possible ASR confusion using the ASR posterior-based loss, while the standard CE loss only focuses on single reference transcripts.

This work is motivated by the work of Osamura et al. [6], which proposed robust cascade ST using ASR word posterior distributions as input for considering ASR output confusion. Our work extends this into the recent multi-task end-to-end ST framework. From another perspective, Chuang et al. [7] employed ASR loss using cosine similarity to consider semantic similarity. The use of distributional loss function in the proposed method is motivated by that work, but our work focuses on the ASR confusion due to pronunciation similarity.

Our proposed method can also be regarded as knowledge distillation [8] using a pre-trained ASR model. Liu et al. [9] employed knowledge distillation in ST using a pre-trained MT model as a *teacher*. Gaido et al. [10] employed knowledge distillation in ST using a pre-trained MT model and ASR model with a loss function based on connectionist temporal classification (CTC) [11] in multi-task learning. Our work does not rely on an additional dataset and focuses on the ASR subtask using ASR posterior distributions to include possible ASR confusions into the ST training.

Experimental results in Fisher Spanish-to-English show that the proposed method resulted in better BLEU scores than the baseline with the standard CE loss with label smoothing.

2. End-to-end Speech Translation

2.1. Single-task End-to-end Speech Translation

An end-to-end ST model consists of a source language speech encoder and a target language text decoder. Let $\mathbf{X} = (x_1, ..., x_T)$ be a source speech feature sequence and $\mathbf{Y} = (y_1, ..., y_N)$ be a target language sequence. Here, T is the length of the speech frame and N is the length of the target text, usually in the number of characters or subwords. For each element v in the target language vocabulary V, the posterior probability of v at the i-th symbol in $\mathbf{Y}$ can be denoted as:

$$P_{\mathrm{ST}}(y_i = v) = p(v|\mathbf{X}, y_{<i}). \tag{1}$$

Its loss function is defined using cross entropy as:

$$\mathcal{L}_{\mathrm{ST}} = -\sum_{i=1}^{N}\sum_{v\in V}^{V} \delta(v, y_i) \log P_{\mathrm{ST}}(y_i = v), \tag{2}$$

where $\delta(v, y_i)$ is an indicator function that returns 1 if $v = y_i$ and otherwise 0. Recent studies usually apply label smoothing to avoid overfit, which distributes the probability mass onto the other elements in V.

2.2. Multi-task End-to-end Speech Translation

Single-task end-to-end ST does not have an explicit ASR module and cannot include any teacher signals for ASR. The multi-task approach introduces another decoder for ASR using the

http://dx.doi.org/10.21437/Interspeech.2021-1105

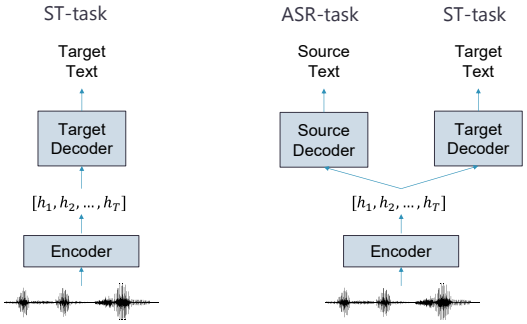


Figure 1: *Single-task ST (left) and multi-task ST (right).*

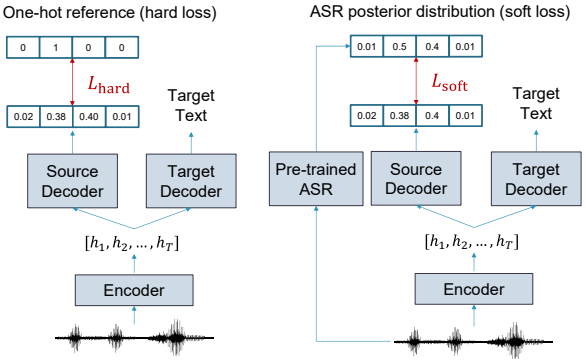


Figure 2: *Hard loss (left) and soft loss (right).*

hidden vectors given by the speech encoder. The ASR-task can be seen as a subtask for the ST problem and its loss function is defined as:

$$\mathcal{L}_{\text{ASR}} = -\sum_{i=1}^{N}\sum_{v\in V}\delta(v, y_i)\log P_{\text{ASR}}(y_i = v). \quad (3)$$

The overall loss function is a weighted sum of the two loss functions $\mathcal{L}_{\text{ST}}$ and $\mathcal{L}_{\text{ASR}}$ defined as follows using a weight λ_{ASR}:

$$\mathcal{L} = (1-\lambda_{\text{ASR}})\mathcal{L}_{\text{ST}} + \lambda_{\text{ASR}}\mathcal{L}_{\text{ASR}}. \quad (4)$$

We illustrate the single-task ST and multi-task ST in Fig. 1. In later sections, we refer to the ASR loss $\mathcal{L}_{\text{ASR}}$ as $\mathcal{L}_{\text{hard}}$ to differentiate it from the proposed one.

3. Proposed Method

3.1. ASR Posterior-based Loss

We propose a method to train an ST model in a multi-task manner using a loss function based on posterior distributions given by a pre-trained ASR model instead of single reference tokens from the gold standard transcripts. Fig. 2 illustrates the difference between a standard *hard* loss calculation and the proposed *soft* one. Here, the ASR decoder should provide large probabilities to hypotheses with similar pronunciations, in practice. In contrast, the standard CE loss with single reference tokens, namely *hard* loss, does not take such situations into account.

The proposed loss function is defined over ASR posterior distributions as references to include the ASR confusion into the hidden vectors in the ST model. The proposed training encourages the model to obtain posterior distributions similar to the pre-trained model in the ASR subtask, as in word-level knowledge distillation [8]. As a result, the ST decoder has to handle the ASR confusion in its training and is expected to be more robust against possible implicit ASR errors in the end-to-end ST.

The posterior distributions are given by a pre-trained ASR model as the outputs from the softmax calculation. Let us define the posterior probability of an ASR token hypothesis v at the i-th position of the ASR result as $P_{\text{soft}}(i, v)$. The $\mathcal{L}_{\text{soft}}$ is defined as:

$$\mathcal{L}_{\text{soft}} = -\sum_{i=1}^{N}\sum_{v\in V}P_{\text{soft}}(i, v)\log P_{\text{ASR}}(y_i = v). \quad (5)$$

Note that P_{ASR} is obtained from the ASR decoder in the ST model and differs from P_{soft} from the pre-trained ASR model.

Table 1: *WERs by our pre-trained ASR model. WERs reported in ESPnet document are also shown for reference.*

Model	dev	dev2	test
Our Model	30.2	29.1	27.2
ESPnet	24.2	23.6	22.8

Finally, we define the overall ASR loss as our proposed *ASR posterior-based loss* which is a weighted mixture of the hard and soft losses as follows:

$$\mathcal{L}_{\text{ASR}} = (1-\lambda_{\text{soft}})\mathcal{L}_{\text{hard}} + \lambda_{\text{soft}}\mathcal{L}_{\text{soft}}, \quad (6)$$

where λ_{soft} is a weight to control their contributions in the training. Here, $\lambda_{\text{soft}} = 0.0$ means that the proposed loss function becomes equivalent to the general CE loss.

4. Experimental Setup

We conducted the following experiments to investigate the effectiveness of the ASR posterior-based loss. We used the Fisher Spanish corpus [12]. This corpus consists of approximately 140K pairs, 160 hours of conversational Spanish speech along with its transcriptions and corresponding English translations. We used it for both ASR and ST model training and developed Spanish-English (Es-En) speech-to-text translation models.

For data pre-processing in all languages, we lowercased and normalized punctuation, followed by tokenization with the `tokenizer.perl` script in the Moses toolkit[1] [13]. We used 80-channel log-Mel filterbank coefficients with 3-dimensional pitch features using Kaldi [14], resulting in 83-dimensional features per frame. The features were normalized by the mean and standard deviation for each training set. We augmented speech data by a factor of 3 by speed perturbation [15]. We removed utterances having more than 3000 frames or 400 characters.

We used subwords for text segmentation for both Spanish and English based on SentencePiece [16], with a shared subword vocabulary with a maximum 1000 entries. The subword model was trained using the training data and applied to all text data. The ST and pre-trained ASR models were based on Transformer [17] and implemented using ESPnet[2] [18]. We used a single random seed of 1 in training.

Finally, we applied model averaging with the best five models among 30 training epochs according to BLEU [19] in the Fisher dev (3.9k pairs) set using beam search with

[1] `https://github.com/moses-smt/mosesdecoder`
[2] `https://github.com/espnet/espnet`

Table 2: *Results measured in BLEU on Fisher with the hyperparameters λ_{ASR} and λ_{soft} resulting the best on dev.*[4]

Model				BLEU		
Task	ASR-task loss	λ_{ASR}	λ_{soft}	dev	dev2	test
Single-task ST	-	-	-	41.10	41.61	40.66
Multi-task ST	CE	0.4	-	44.50	46.20	44.88
	CE + Label smoothing	0.5	-	45.29	46.34	45.16
	ASR Posterior-based loss (Proposed)	0.4	0.5	**45.54**	**46.46**	**45.64**

Table 3: *Results measured in BLEU on Fisher with different λ_{ASR}. λ_{soft} were the ones resulting the best on dev.*

Model				BLEU		
Task	ASR-task loss	λ_{ASR}	λ_{soft}	dev	dev2	test
Single-task ST	-	-	-	41.10	41.61	40.66
Multi-task ST	CE	0.3	-	44.08	45.07	**44.69**
	CE + Label smoothing		-	44.23	45.25	44.21
	ASR Posterior-based loss (Proposed)		0.9	**44.99**	**45.73**	44.66
	CE	0.4	-	44.50	46.20	44.88
	CE + Label smoothing		-	44.61	45.30	45.01
	ASR Posterior-based loss (Proposed)		0.5	**45.54**	**46.46**	**45.64**
	CE	0.5	-	43.64	45.28	43.83
	CE + Label smoothing		-	45.29	46.34	45.16
	ASR Posterior-based loss (Proposed)		0.5	**45.46**	**46.37**	**46.04**

the beam size of 10. The averaged model was applied for Fisher dev2 (3.9k pairs) and test (3.6k pairs) data for evaluation. We evaluated on 4-references case-insensitive BLEU with `multi-bleu.detok.perl` in Moses.

4.1. Pre-trained ASR Model

The pre-trained ASR model was trained using the pairs of Spanish speech and transcripts in the training data. The hyperparameters of the model almost followed the default settings of ESPnet. The encoder was a 12-layer 2048-dimensional transformer with 6-head 256-dimensional attention with a 6-layer 2048-dimensional decoder. The label smoothing weight for CE loss was 0.1. Batch size was 64 and accumgrad size was 2. We chose the best parameter set according to the subword accuracy in the development set from among 30 training epochs. Table 1 shows the performance in word error rate (WER) of the pre-trained ASR model using beam search with a beam size of 10. We used this pre-trained ASR model to generate teacher distributions with greedy search for time efficiency in the decoding of the whole training set. Table 1 also shows the performance mentioned in the ESPnet document[3]. We can see that the performance of our model is worse. The reason would be the difference in the model configuration; the reported results were from the model trained using CTC-based loss.

4.2. ST Model

The ST model was configured almost the same as the ASR model above. One major difference was the existence of two decoders for the ST maintask and the ASR subtask. We applied label smoothing for the ST-task with a weight of 0.1. We set the batch size to 64 and accumgrad size to 4.

[3] `https://github.com/espnet/espnet/blob/master/egs/fisher_callhome_spanish/asr1b/RESULTS.md`

4.3. Baseline Models

We compared the proposed method with two baselines: a basic multi-task ST with standard CE loss and that applying label smoothing with a weight of 0.1.

4.4. Proposed Models[5]

We chose hyperparameter values according to BLEU in the development set. We chose λ_{ASR} in the equation 4 from among $\{0.3, 0.4, 0.5\}$ and chose λ_{soft} in the equation 6 from among $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$.

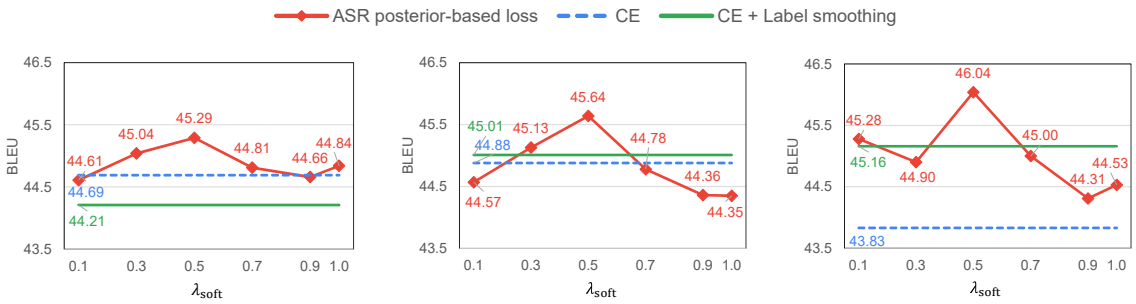
5. Results and Discussion

Table 2 shows the performance of the ST models in BLEU scores with the best hyperparameter values in the dev set. The results clearly show that the proposed method outperformed the baselines. Table 3 shows the performance of the ST models for different λ_{ASR}. The results also show most of the proposed models outperformed the baselines.

For a detailed analysis, Fig. 3 shows the BLEU results on the Fisher test set on different λ_{soft} in each λ_{ASR}. The proposed method achieved the best BLEU scores with $\lambda_{\text{soft}} = 0.5$ and showed degradation with the other weights. This was also observed for the Fisher dev2 results, which are excluded from the figure for simplicity. One possible reason for the degradation is the performance of the pre-trained ASR model. It was not good enough to fully depend on the soft loss, so the mixed use of soft and hard loss was important. This also suggests that the proposed method works effectively even when the pre-trained ASR is not good enough. There are differences in the loss values in the training time. Fig. 4 shows the validation loss values with

[4] Our multi-task ST performance has some gaps between ASR-task multi-task ST model in ESPnet [20] with CTC-based loss. We focus on using the attention-based loss calculation method and apply it to our proposed method.

[5] Our code is available at: `https://github.com/ahclab/st-asrpbl`



(a) $\lambda_{\text{ASR}} = 0.3$ (b) $\lambda_{\text{ASR}} = 0.4$ (c) $\lambda_{\text{ASR}} = 0.5$

Figure 3: *Fisher test BLEU on different λ_{soft} in each λ_{ASR}.*

(a) $\lambda_{\text{soft}} = 0.0$ (CE) (b) $\lambda_{\text{soft}} = 0.3$ (c) $\lambda_{\text{soft}} = 0.5$ (d) $\lambda_{\text{soft}} = 0.7$ (e) $\lambda_{\text{soft}} = 1.0$

Figure 4: *Validation loss on each λ_{soft} in $\lambda_{ASR} = 0.4$.*

$\lambda_{\text{ASR}} = 0.4$ and varying λ_{soft}. The total validation loss became larger with larger λ_{soft}. This suggests that the proposed method introduced larger ASR confusion in the training that might have worked as regularization. From Fig. 3 and Fig. 4, giving a too large weight on the soft loss degraded translation quality, but the appropriate mixture with the hard loss was effective for ST.

For analysis of output sentences between the baseline and proposed model, Table 4 shows some translation examples given by the models of the baseline CE loss with label smoothing and proposed ASR posterior-based loss shown in Table 2. We didn't have ASR results from ASR-task in this paper. From Example 1, in the baseline model, the word *relationships* appears instead of the reference word *relaxation*. This would be due to ASR confusion with the Spanish word *relaciones*. On the other hand, the proposed method predicted the correct word. It appears that the model is more robust for ASR confusion. From Example 2, *sobrino* (Spanish for *nephew*) would be translated to *nephew* in English also. *Sobrino* is very similar to *sobrina* in Spanish. As a result, the ASR-task on the baseline model predicted the incorrect word *sobrino* for the correct word *sobrina*, and the translation was also affected by the error. In contrast, the proposed model gave the correct translation. The proposed method enabled robust ST for ASR confusion using ASR posterior-based loss.

6. Conclusions

In this paper, we proposed ASR posterior-based loss to handle ASR confusion in multi-task end-to-end ST. The proposed loss function works as knowledge distillation from a pre-trained ASR model and encourages robust ST against ASR confusion.

Table 4: *Fisher test examples in Fig. 2 settings.*

Example 1	
Ground Truth (Es)	para relajación
Ground Truth (En)	for relaxation
CE + Label smoothing (En)	for relationships
ASR Posterior-based loss (En)	for **relaxing**

Example 2	
Ground Truth (Es)	quién no su sobrina
Ground Truth (En)	who no your niece
CE + Label smoothing (En)	who his nephews
ASR Posterior-based loss (En)	who are your **nieces**

Our experimental results showed the effectiveness of the proposed method compared with the baselines with standard CE loss with label smoothing. The results also suggest that mixed use of the proposed and standard loss is important rather than using either one of them.

Future work includes further investigation with different pre-trained ASR models and other language pairs. The extension to using phonetic information like Salesky et al. [21] in loss calculation for making robust ST against acoustic similar tokens is also a promising future direction.

7. Acknowledgements

Part of this work is supported by JSPS KAKENHI Grant Numbers JP17H06101.

8. References

[1] M. Sperber, G. Neubig, N. Pham, and A. Waibel, "Self-Attentional Models for Lattice Inputs," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 1185–1197.

[2] A. Anastasopoulos and D. Chiang, "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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 82–91.

[3] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct Speech-to-Speech Translation with a Sequence-to-Sequence Model," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1123–1127.

[4] T. Kano, S. Sakti, and S. Nakamura, "End-to-End Speech Translation With Transcoding by Multi-Task Learning for Distant Language Pairs," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1342–1355, 2020.

[5] T. Kano, S. Sakti, and S. Nakamura, "Transformer-Based Direct Speech-To-Speech Translation with Transcoder," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 958–965.

[6] K. Osamura, T. Kano, S. Sakti, K. Sudoh, and S. Nakamura, "Using Spoken Word Posterior Features in Neural Machine Translation," *Proceedings of the 15th International Workshop on Spoken Language Translation, 181-188, Oct. 2018*, vol. 21, p. 22, 2018.

[7] S. Chuang, T. Sung, A. H. Liu, and H. Lee, "Worse WER, but Better BLEU? Leveraging Word Embedding as Intermediate in Multitask End-to-End Speech Translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 5998–6003.

[8] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[9] Y. Liu, H. Xiong, J. Zhang, Z. He, H. Wu, H. Wang, and C. Zong, "End-to-End Speech Translation with Knowledge Distillation," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1128–1132.

[10] M. Gaido, M. A. D. Gangi, M. Negri, and M. Turchi, "End-to-End Speech-Translation with Knowledge Distillation: FBK@IWSLT2020," in *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, M. Federico, A. Waibel, K. Knight, S. Nakamura, H. Ney, J. Niehues, S. Stüker, D. Wu, J. Mariani, and F. Yvon, Eds. Association for Computational Linguistics, 2020, pp. 80–88.

[11] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, ser. ACM International Conference Proceeding Series, W. W. Cohen and A. W. Moore, Eds., vol. 148. ACM, 2006, pp. 369–376.

[12] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association, 2004.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, J. A. Carroll, A. van den Bosch, and A. Zaenen, Eds. The Association for Computational Linguistics, 2007.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[15] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 3586–3589.

[16] T. Kudo and J. Richardson, "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*, 2018, pp. 66–71.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-End Speech Processing Toolkit," *Proc. Interspeech 2018*, pp. 2207–2211, 2018.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[20] H. Inaguma, S. Kiyono, K. Duh, S. Karita, N. Yalta, T. Hayashi, and S. Watanabe, "ESPnet-ST: All-in-One Speech Translation Toolkit," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, A. Celikyilmaz and T. Wen, Eds. Association for Computational Linguistics, 2020, pp. 302–311.

[21] E. Salesky and A. W. Black, "Phone Features Improve Speech Translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 2388–2397.