ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414159

JOINTLY TRAINED TRANSFORMERS MODELS FOR SPOKEN LANGUAGE TRANSLATION

Hari Krishna Vydana, Martin Karafiát, Katerina Zmolikova, Lukáš Burget, "Honza" Černocký

Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia

ABSTRACT

End-to-End and cascade (ASR-MT) spoken language translation (SLT) systems are reaching comparable performances, however, a large degradation is observed when translating the ASR hypothesis in comparison to using oracle input text. In this work, degradation in performance is reduced by creating an End-to-End differentiable pipeline between the ASR and MT systems. In this work, we train SLT systems with ASR objective as an auxiliary loss and both the networks are connected through the neural hidden representations. This training has an End-to-End differentiable path with respect to the final objective function and utilizes the ASR objective for better optimization. This architecture has improved the BLEU score from 41.21 to 44.69. Ensembling the proposed architecture with independently trained ASR and MT systems further improved the BLEU score from 44.69 to 46.9. All the experiments are reported on English-Portuguese speech translation task using the How2 corpus. The final BLEU score is on-par with the best speech translation system on How2 dataset without using any additional training data and language model and using fewer parameters.

Index Terms: Spoken Language Translation, Transformers, Joint training, How2 dataset, Auxiliary loss, ASR objective, Coupled decoding, End-to-End differentiable pipeline.

1. INTRODUCTION

Spoken Language Translation (SLT) refers to the task of transcribing a spoken utterance in a source language into the target language. SLT systems are typically categorized into the cascade and End-to-End systems. Cascade SLT systems in their popular form comprise an automatic speech recognition (ASR) system followed by a machine translation (MT) system. The initial ASR system generates the text sequence for the spoken utterance and the generated text sequence is translated into the target language by an MT system. Speech recognition task has a monotonic alignment between the spoken sequence and the labeling sequence. But this property is not present in MT and SLT tasks. ASR system relies more on the local context to transcribe the speech while MT systems need the wider context of a word to translate it. In a cascade approach, both ASR and MT are trained separately. The improvements in both ASR and MT performance can be easily translated to the SLT model. End-to-End SLT system is a single model that directly transcribes the spoken utterance in the target language. End-to-End SLT models have to learn the complex mapping between the spoken sequence and the labeling sequence in the target language, which involves the word splitting, merging, and reordering. End-to-End SLT systems can result in simpler models to train, have low latency and lead to direct optimization of the required task. The difference between the cascade and the End-to-End system is that the End-to-End system does not use the source text for translation. However, while optimizing the End-to-End SLT model, the use of the source text has shown to improve the performance [1].

Currently, cascade SLT systems perform better than the End-to-End SLT systems. Popular End-to-End SLT systems are sequence-to-sequence with attention models. With the recent progress in these models, the performance gap between cascade and End-to-End SLT systems is narrowing. Nevertheless, the cascade SLT system suffers from a large performance gap when MT models are used to translate ASR hypotheses compared to using oracle input text. This large performance gap is due to the erroneous ASR hypotheses and the MT system which is not able to handle the errors. In this study, we explore approaches to reduce this performance gap by creating an End-to-End differentiable pipeline between ASR and MT systems. SLT systems are trained with ASR objective as an auxiliary loss and both the networks are connected through the neural hidden representations. This way the model has a differentiable path between input (speech) and the final labeling sequence and also utilizes the ASR objective to improve the performance of the SLT system. During the inference, both the models are connected through neural hidden representations of N-best hypotheses. The coupled search using the log-likelihood scores from both ASR and MT models is used to pick the best translation sequence. During the training, the proposed model is similar to the model described in [2], where the context vectors from the ASR decoder are passed as input to the MT model. During the inference, the proposed model is similar to [3], where back translation likelihoods are used to re-rank the hypotheses, proposed model uses the likelihoods from the ASR decoder to re-rank the MT hypotheses.

1.1. Recent Developments in SLT systems

Here, we describe some of the recent SLT models submitted to IWSLT-2019. The evaluation had a track on English to Portuguese translation on the same development set as used in our experiments. The best performance in IWSLT 2019 [4] was achieved with a cascade SLT system with a pipeline of ASR, punctuation model and MT systems. The ASR used in that system is an ensemble of LSTM-based encoder-decoder model and a large transformer model($>$150M parameters) trained with stochastic depth [5]. The MT model used is a multi-lingual model ($>$200M parameters) trained for two languages (English ⟶ German and English ⟶ Portuguese). Transformer models are adapted for training End-to-End SLT systems with an additional distance penalty for the attention to focus more on the local context [6]. Data augmentation methods such as spec-augment [7] and the use of back-translated synthetic text has improved the performance [7]. The End-to-End SLT systems with their parameters initialized from independently trained ASR and MT systems have been studied in [8]. LSTM-encoder-decoder models have been used to train End-to-End SLT systems with characters as target units in [9]. Multi-model SLT systems, which have used the features from text and video were studied in [10].

2. DATABASE

All the experiments in this paper are conducted using HOW2 data-set [11]. The data-set comprises train, dev, and test sets. The training, development and test sets consist of 185K, 2305, and 2022 sentences, which amounts to 298, 3, and 4 hours of speech data. All the sentences in the data-set have parallel Portuguese translations. All the models in this work are trained using the training set and early stopping is controlled using the development set. All the text is lower-cased and the punctuation symbols are removed. The sentence piece model is trained to obtain a sub-word vocabulary of 5000 tokens for both English and Portuguese texts. Audio from the videos is extracted and a 16 kHz signal is used in these experiments. 40-Dimensional Mel-filter bank features are extracted with 25 ms window size and 10 ms overlap. Mean and variances of the features are normalized per-video. The performances of all the translation systems are measured using Sacre-BLEU. The results presented in this paper are compared with the baseline results presented in [11].

3. CASCADE VS. END-TO-END SLT SYSTEMS

3.1. Transformer ASR systems

Similar to [12, 13, 14], our transformer ASR models are trained with characters or Byte Pair Encoding (BPE) units as target units. In both cases, the models are trained with 12 encoder layers and 6 decoder layers. The sizes of the hidden and feed-forward layers are 256 and 1024, respectively. The models are trained for 150 epochs. Each batch comprised 7000 target units. The models from the last ten epochs are averaged and the averaged weights are used during the decoding. ASR hypotheses are decoded with a beam size of 10. The start-of-the-sequence(<SOS>) and end-of-the-sequence (<EOS>) are modeled by additional tokens. The data-set has some very long sequences with 400-500 characters and decoding these sentences increases the decoding time. To reduce the decoding time, a vectorized beam search described in [15] has been used. A threshold mechanism described in [16] has been used to pick the proper EOS candidates. The length penalties of 1 and 0.8 are used for character and BPE models. No external language model has been used in this work. The performance of both models is presented in Table 1. The first row corresponds to an ASR system trained with TDNN-LFMMI using Kaldi-recipes. This ASR is not used in any SLT system but is presented only as a reference. The second row of the table shows the performance of the LSTM-based sequence-to-sequence model presented in [11]. Rows 3 and 4 show the performances of ASR systems based on transformer models with characters and BPE (5K) units as the target units. The transformer model with BPE targets performs better.

Table 1. ASR systems trained using Transformer Models.

Architecture (Target units)	Dev set(WER)	Test set(WER)
TDNN-LFMMI(BPE)	-	13.7
S2S-Attention (BPE) [11]	-	19.4
Transformer (character)	17.47	17.87
Transformer (BPE)	16.85	16.52

3.2. Transformer based Machine Translation systems

Transformer models have been proposed for Machine Translation (MT) in [12]. Our MT systems are trained with three different input-output granularities such as characters-characters, characters-BPE, and BPE-BPE. The model has 6-encoder layers, 6-decoder layers. The sizes of the hidden and feed-forward layers are 512 and 1024, respectively. The models have 8 parallel heads and are trained as described in section 3.1. The beam size and length penalties for decoding are tuned on the development set. The optimal beam size for all the granularities is 5, and the length penalties for characters-characters, characters-BPE, and BPE-BPE are 1.2, 1.0, 1.2 respectively. The tokens predicted from the model are converted back to the text and the Sacre-BLEU is computed. The EOS thresholding described in section 3.1 has been used. The performances of the trained MT systems are presented in Table 2. From Table 2, it can be observed that both character-character and BPE-BPE based models have comparable performances. Empirically, it has been observed that the character-character based MT models have taken longer time to train and decode compared to other models. The models with character input and BPE as output has performed better than the other two systems.

Table 2. MT systems trained using Transformer Models.

Architecture (Input-Output) Granularity	Dev Sacre-BLEU	Test Sacre-BLEU
S2S-Attention-(BPE) [11]		54.4
Transformer-(characters-characters)	53.08	52.08
Transformer-(BPE-BPE)	53.16	52.01
Transformer-(characters-BPE)	55.32	54.80

3.3. Cascade SLT models

In this work, two different cascade systems have been trained: ASR systems with either characters or BPE output units along with the corresponding MT model. The ASR and MT models described in sections 3.1 and 3.2 are used in the cascade pipeline. The ASR hypothesis is decoded and the decoded hypothesis is used by the MT model to produce the translation. The performance of the cascade SLT systems is presented in Table 3. We can see that the performance of the SLT systems is significantly worse than the performance of the MT systems shown in Table 2. Due to the use of the erroneous ASR hypothesis as input to the MT system, the performance of the MT systems has degraded in the BLEU score by more than 10 points. Columns 2 and 3 of Table 3 show the performance of SLT systems using the 1-best hypothesis. Columns 4 and 5 show the performance of SLT systems using the n-best hypothesis as de-

Table 3. Cascade SLT models (ASR-MT) trained using Transformer. The performances in the below table are presented in-terms Sacre-BLEU scores.

Architecture	1-best		n-best	
(Input-Output) Granularity	Dev set	Test set	Dev set	Test set
Transformer (characters-characters)	39.12	39.18	-	-
Transformer (characters-BPE)	41.21	41.31	42.52	42.31
Transformer (BPE-BPE)	41.86	41.71	43.68	43.6

fined in equation (1), i.e. the sum of log-likelihoods of ASR and MT model are used to pick the best translation.

$$\begin{aligned} P(y|x) &= \arg\max_{y} \sum_{z} P(y|z)P(z|x)) \\ &\approx \arg \max_{y \in \hat{Y}(z), z \in \hat{Z}(x)} logP(y|z) + logP(z|x) \end{aligned} \tag{1}$$

In equation (1), x is the source speech, y is the target token sequence and z is the source token sequence and $\hat{Z}$, $\hat{Y}$ are the n-best hypotheses from ASR and MT models. As we can see from the results, the n-best approach improves the performance.

3.4. End-to-End SLT models

Our End-to-End SLT systems are again based on the transformer architecture [6, 17]. The models predict the Portuguese characters/BPE units. The training and the decoding follow the description from section 3.1. The performances of the End-to-End SLT systems are tabulated in Table 4. We have experimented with a different number of the encoder (enc) and decoder (dec) layers as indicated in the first column of the table. Unless stated otherwise, the models have 4 heads and the hidden and feed-forward layer sizes 256 and 1024, respectively. From the table, we can see that the models with BPE target units perform better than the models with character targets. However, the performance of these models is worse than the performance of the cascade SLT models. The performance improves as we add more parameters to the models. The last line of Table 4 corresponds to a large transformer model with 20 encoder and 10 decoder layers and hidden layer size 512.

Table 4. Transformer based End-to-End SLT.

Architecture (Target unit) (No.of parameters)	Dev set	Test set
S2S-Attention-BPE [18]	-	36.0
Transformer-12enc-6dec-(char)(24M)	33.4	33.8
Transformer-12enc-8dec-(char)(29M)	33.97	33.98
Transformer-8enc-6dec-(BPE)(24M	36.91	37.45
Transformer-12enc-6dec-(BPE)(30M)	38.95	39.06
Transformer-20enc-10dec-(BPE)(46M)	40.03	40.19
Transformer-20enc-10dec-$d_{hidden-size}$-512(BPE)(92M)	39.94	40.59

3.5. Augmented Training for Cascade SLT

To reduce the mismatch between the ASR hypothesis and the oracle text, the text training corpus for the MT systems is augmented with the ASR hypothesis. ASR hypothesis for the training data is obtained using the ASR model described in section 3.1. The performance of these models is presented in Table 5. The different rows correspond to the different cascade SLT systems with different input/output granularities. Columns 2 and 3 show performances of the cascade SLT systems trained with the augmented inputs and evaluated with the oracle input text. Columns 4 and 5 show performances of the systems evaluated with the ASR hypothesis. Comparing the results in Table 5 with 1-best results in Table 3 the systems trained with augmented data have not significantly improved the performance when evaluated on ASR hypotheses and perform worse with the oracle input text. The models trained from scratch with the augmented data and the models that are initially trained on clean data and later fine-tuned for augmented data have yielded similar performances.

Table 5. Cascade SLT systems trained using augmented data.

	Oracle Text		ASR hypothesis	
	Dev set	Test set	Dev set	Test set
Transformer-(char-char)	50.21	49.44	39.67	39.54
Transformer-(char-BPE)	54.2	53.98	41.04	41.30
Transformer-(BPE-BPE)	49.59	49.41	39.12	39.82

4. MULTI-TASK TRAINING OF SLT SYSTEMS WITH ASR OBJECTIVE AS AN AUXILIARY LOSS

From the above sections, it can be observed that the cascade SLT systems perform better than the End-to-End approaches. At the same time, there is a large performance degradation when MT systems translate ASR hypothesis as opposed to using oracle input text. To reduce this performance gap, we have trained a model with an End-to-End differentiable pipeline between the spoken sequence and the target token sequence. This architecture uses the ASR objective as an auxiliary loss. The architectures for ASR and MT models described in the above sections are connected as shown in the right block of Figure 1. The pre-softmax activations are taken from the ASR as continuous hidden representations and they are used as the input to train the MT model. This way MT does not rely on discrete decisions made by the ASR and the whole pipeline can be (also) optimized for the final MT objective. We optimize the combined model with a multi-task loss function. The ASR model parameters in the pipeline are optimized for both ASR and MT objective, while the MT model parameters are optimized for MT objective only. The models are trained as described in section 3.1. The model has two decoders in the pipeline: The ASR decoder produces the 10-best ASR hypothesis and the corresponding hidden representations. The hidden representation from each of the 10-best outputs is considered as a separate input to MT. For each such input, the MT decoder produces the 5-best MT hypothesis. All 50 hypotheses produced by the MT model are combined and the best hypothesis from the MT model is used as the output hypothesis.

Table 6. SLT systems jointly trained with a Multi-Task objective and ASR as an auxiliary loss

	Sacre-BLEU		WER	
	Dev set	Test set	Dev set	Test set
Transformer-End-to-End	36.2	36.8	-	-
Transformer-(char-BPE)	40.48	40.46	24.27	24.50
Transformer-(BPE-BPE)	44.9	44.18	17.99	18.09

The performances of the proposed SLT systems are tabulated in Table 6. We can see that, with the BPE ASR target units, the multi-task training of the whole pipeline outperforms the corresponding cascade SLT system (compare with column 4 and 5 in Table 3). The proposed multi-task training improves the BLEU score by around 1 point compared to the cascade system with n-best hypotheses and by 4 around or 5 points compared to End-to-End systems from Table 4.

Unfortunately, with the character ASR target units, the multi-task training degrades the performance as compared to the corresponding cascade system. In this case, the ASR module of the joint pipeline has significantly worse ASR performance (see WER in columns 4 and 5 in Table 6). As a consequence, the MT performance also degrades. In this case, the difficulty with training the

ASR module is likely caused by the mismatch in the granularity of the ASR and MT output representation (i.e. characters vs. BPEs).

5. ENSEMBLING WITH EXTERNAL ASR AND MT SYSTEMS

From Table 6, it can be observed that the WER performance of the ASR module in the joint training is not on-par with the performance of ASR systems trained only for the ASR objective described in subsection 3.1. To improve this performance the ASR module is ensembled with the external ASR model described in subsection 3.1. During the inference, the softmax output distributions from both models are computed for each prefix and the distributions are averaged and this average distribution is used for the beam search.

In a similar way, we also ensemble the MT module: The proposed jointly trained pipeline also produces characters/BPE units as outputs along with the neural-hidden representations, which can also be used for translation with the external MT models described in section 3.2. The left block of Figure 1 depicts the ensembling with an external MT system. In our experiments, both the ASR and MT independently trained external models described in subsection 3.1 and 3.2 are ensembled with their corresponding modules of the joint model.

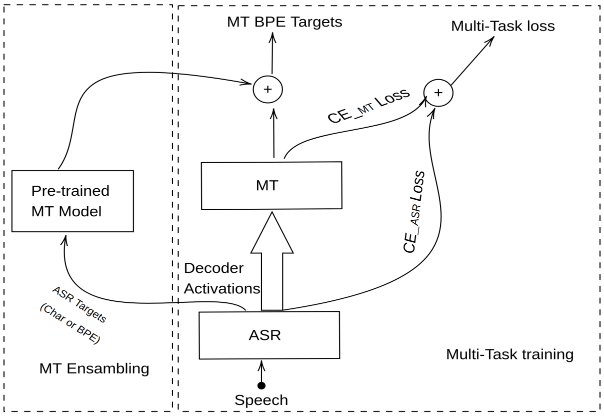

Fig. 1. Block diagram describing the Multi-Task training with ASR objective as an auxiliary loss and MT ensembling.

Table 7. Multi-Task training with ASR as a auxiliary loss and ensembling with external ASR and MT systems.The performances in the below table are presented in-terms Sacre-BLEU scores.

	ASR-Guided		Joint-model+ Ensemble-ASR		Joint-model+ Ensemble-MT		Joint-model+ Ensemble-ASR+MT	
	Dev set	Test set	Dev set	Test set	Dev set	Test set	Dev set	Test set
Transformer- (char-BPE)	40.84	40.81	41.92	41.70	43.04	42.31	**43.72**	**43.06**
Transformer- (BPE-BPE)	45.32	44.69	46.93	46.22	45.8	45.7	**47.33**	**46.9**

The performances obtained with different ensembling configurations are tabulated in Table 7. Columns 2 and 3 show the performances of the ASR-Guided jointly trained model. In this case, the n-best sequence is obtained from an independently trained ASR system described in section 3.1. The n-best ASR hypotheses are rescored by (passed through a single decoder iteration of) the joint model to obtain the neural hidden representations, which are then used by the MT-module of the joint model to produce the translations. Column 4, 5, and 6, 7 are the performances of SLT systems with ASR and MT ensembling, respectively. Columns 8 and 9 show the performances of SLT systems with both ASR and MT ensembling. We can see that the larger improvements in BLEU score are obtained with MT ensembling than with ASR ensembling. These improvements could be attributed to the higher diversity of the MT input representations (MT module in joint pipeline uses the continuous neural representation, while the external MT model uses discrete representations) and also to the fact that the MT model is at the end of the pipeline. The best results are obtained with ensembling ASR and MT giving the best BLEU-score of 47.33 and 46.9 on development and test sets of HOW2 data-set. These results are on-par with the best performing systems on How2 data-set published in IWSLT-2019 [4], [1]. The total number of parameters in the model (ASR+MT) is around 140M, which is much lesser than (>350M parameters) for the models in [4].

Note, that the ensembling with independently trained external ASR and MT models could be especially beneficial in scenarios with additional available ASR and MT training data that are not a paired speech-translated text data. The performance of the proposed model is evaluated on IWSLT-corpus and the results are reported in [1]

6. CONCLUSION & FUTURE SCOPE

Large performance degradation is observed while translating the ASR hypothesis as opposed to using oracle input text. Proposed systems aim to reduce this degradation by training models with an end-to-end differentiable pipeline between ASR and MT models. Introduction of the ASR objective as an auxiliary loss while optimizing the combined ASR+MT models has improved the performance of SLT systems. The performance gains are higher when both ASR and MT models use the target units of the same granularity (BPE in our case). As all the models are transformers, they could be replaced with Non-Auto regressive models [19], [20],[21], which could reduce the latency of decoding. Along with the input symbols, a mechanism to present the confidence of the input symbol could help MT models to better translate the ASR hypothesis. This could be an interesting direction for further study. While training with the erroneous input text, using a sentence-wise confidence metric and conditioning the learning of the model on the confidence metric could improve the performance of MT [22]. Using unpaired additional data for training ASR and MT models could improve the performance of the ensemble.

7. ACKNOWLEDGMENTS

The work was supported by Czech National Science Foundation (GACR) project "NEUREM3" No. 19-26934X, and by European Union's Horizon 2020 projects No. 864702 - ATCO2. It was also supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) MATERIAL program, via Air Force Research Laboratory (AFRL) contract # FA8650-17-C-9118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government.

[1] https://arxiv.org/submit/3430124/preview

8. REFERENCES

[1] Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Enrique Yalta Soplin, Tomoki Hayashi, and Shinji Watanabe, "Espnet-st: All-in-one speech translation toolkit," 2020.

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

[3] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li, "Neural machine translation with reconstruction," *arXiv preprint arXiv:1611.01874*, 2016.

[4] Thanh-Le Ha Juan Hussain Felix Schneider Jan Niehues Sebastian Stüker Alexander Waibel Ngoc-Quan Pham, Thai-Son Nguyen, "The iwslt 2019 kit speech translation system," Nov. 2019.

[5] Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Muller, and Alex Waibel, "Very deep self-attention networks for end-to-end speech recognition," *arXiv preprint arXiv:1904.13377*, 2019.

[6] Mattia A Di Gangi, Matteo Negri, and Marco Turchi, "Adapting transformer to end-to-end spoken language translation," pp. 1133–1137, 2019.

[7] M Di Gangi, Matteo Negri, Viet Nhat Nguyen, Amirhossein Tebbifakhr, and Marco Turchi, "Data augmentation for end-to-end speech translation," 2019.

[8] Hirofumi Inaguma, Xuan Zhang, Zhiqi Wang, Adithya Renduchintala, Shinji Watanabe, and Kevin Duh, "The jhu/kyotou speech translation system for iwslt 2018," in *Proc. The International Conference on Spoken Language Translation*. IWSLT, 2018.

[9] Ha Nguyen, Natalia Tomashenko, Marcely Zanon Boito, Antoine Caubriere, Fethi Bougares, Mickael Rouvier, Laurent Besacier, and Yannick Esteve, "On-trac consortium end-to-end speech translation systems for the iwslt 2019 shared task," 2019.

[10] Zixiu Wu, Ozan Caglayan, Julia Ive, Josiah Wang, and Lucia Specia, "Transformer-based cascaded multimodal speech translation," 2019.

[11] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze, "How2: a large-scale dataset for multimodal language understanding," *arXiv preprint arXiv:1811.00347*, 2018.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. IEEE, 2018, pp. 5884–5888.

[14] Shinji Watanabe Marc Delcroix Atsunori Ogawa Tomohiro Nakatani Shigeki Karita, Nelson Enrique Yalta Soplin, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," pp. 1408–1412, 2019.

[15] Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Niko Moritz, and Jonathan Le Roux, "Vectorized beam search for ctc-attention-based speech recognition," in *Proc. INTERSPEECH*, 2019, pp. 3825–3829.

[16] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," *arXiv preprint arXiv:1909.09116*, 2019.

[17] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on transformer vs rnn in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[18] Vikas Raunak, Sang Keun Choe, Quanyang Lu, Yi Xu, and Florian Metze, "On leveraging the visual modality for neural machine translation," *arXiv preprint arXiv:1910.02754*, 2019.

[19] Jindřich Libovickỳ and Jindřich Helcl, "End-to-end non-autoregressive neural machine translation with connectionist temporal classification," *arXiv preprint arXiv:1811.04719*, 2018.

[20] Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition," 2019.

[21] Jason Lee, Elman Mansimov, and Kyunghyun Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," 2018.

[22] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020.