Towards Efficient Simultaneous Speech Translation: CUNI-KIT System for Simultaneous Track at IWSLT 2023

Peter Polák[1] and Danni Liu[2] and Ngoc-Quan Ngoc[2]

Jan Niehues[2] and Alexander Waibel[2,3] and Ondřej Bojar[1]

polak@ufal.mff.cuni.cz

[1] Charles University [2] Karlsruhe Institute of Technology

[3] Carnegie Mellon University

Abstract

In this paper, we describe our submission to the Simultaneous Track at IWSLT 2023. This year, we continue with the successful setup from the last year, however, we adopt the latest methods that further improve the translation quality. Additionally, we propose a novel online policy for attentional encoder-decoder models. The policy prevents the model to generate translation beyond the current speech input by using an auxiliary CTC output layer. We show that the proposed simultaneous policy can be applied to both streaming blockwise models and offline encoder-decoder models. We observe significant improvements in quality (up to 1.1 BLEU) and the computational footprint (up to 45 % relative RTF).

1 Introduction

Simultaneous speech translation (SST) is the task of translating speech into text in a different language before the utterance is finished. The goal of SST is to produce a high-quality translation in real-time while maintaining low latency. However, these two objectives are conflicting. If we decrease the latency, the translation quality also drops. Last year's IWSLT evaluation campaign (Anastasopoulos et al., 2022) showed that current methods for simultaneous speech translation can approach the translation quality of human interpreters (Polák et al., 2022). The disadvantage is a higher computation footprint that might make a widespread application prohibitive.

This paper describes the CUNI-KIT submission to the Simultaneous translation track at IWSLT 2023 (Agarwal et al., 2023). Following our last year's submission (Polák et al., 2022), we continue in our effort to onlinize the robust offline speech translation models. However, the main goal of this submission is to improve the computational footprint. To this end, we propose a novel online policy based on CTC. As we experimentally document, the online CTC policy can be used to onlinize the offline models achieving a 45 % improvement in real time factor (RTF) as well as to improve the quality of the streaming blockwise models (Tsunoo et al., 2021). Aside from improving the online policy, we also adopt the novel improved streaming beam search (Polák et al., 2023) that further improves the translation quality.

Our contributions are as follows:

- We adopt the latest online decoding algorithm that improves the translation quality of robust offline models in the simultaneous regime,
- We propose a novel online policy that significantly
  - lowers the computational complexity of the online decoding with robust offline models while maintaining the same or only slightly worse translation quality,
  - improves the translation quality of the streaming blockwise models while maintaining the same latency,
- We demonstrate that our systems can run on hardware accessible to a wide audience.

2 Methods

In our submission, we use two different model architectures — a traditional offline ST architecture and a blockwise simultaneous ST architecture (Tsunoo et al., 2021). In this section, we describe the methods applied to achieve simultaneous ST using these architectures.

2.1 Incremental Blockwise Beam Search with Controllable Quality-Latency Tradeoff

To use the traditional offline ST model in a simultaneous regime, Liu et al. (2020) proposed chunking, i.e., splitting the audio source utterance into small constant-length chunks that are then incrementally

Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023), pages 389–396
July 13-14, 2023 ©2023 Association for Computational Linguistics

fed into the model. As translation quality tends to diminish toward the end of the unfinished source, an online policy is employed to control the latency-quality tradeoff in the generated output. Popular online policies include wait-k (Ma et al., 2019), shared prefix (Nguyen et al., 2020), hold-n and local agreement (Liu et al., 2020). In Polák et al. (2022), we showed that the tradeoff could be controlled by varying the chunk length.

To generate the translation, a standard beam search is typically applied (Sutskever et al., 2014). While this decoding algorithm enables the model to generate a complete translation for the current input, it also suffers from overgeneration (i.e., hallucinating tokens beyond sounds present in the input segment) and low-quality translations towards the end of the source context (Dong et al., 2020; Polák et al., 2022).

To tackle this issue, we adopt an improved incremental blockwise beam search (Polák et al., 2023). We outline the algorithm in Algorithm 1 and highlight the main differences from the original approach used in Polák et al. (2022) with red.

Algorithm 1: Incremental blockwise streaming beam search algorithm for incremental ST

```
   Input  : A list of blocks, an ST model
   Output : A set of hypotheses and scores
1  Seen ← ∅;
2  for each block do
3      Encode block using the ST model;
4      Stopped ← ∅;
5      minScore ← −∞;
6      while #active beams > 0 and not max. length do
7          Extend beams and compute scores;
8          for each active beam b do
9              if b ends with <eos> or (score ≤ minScore
                 and b ∉ Seen) then
10                 minScore ← max(minScore, score);
11                 Stopped ← Stopped ∪ b;
12                 Remove b from the beam search;
13             end
14         end
15     end
16     Seen ← Seen ∪ Stopped;
17     Sort Stopped by length-normalized score;
18     Set the best hypothesis from Stopped as active beam;
19     Apply the incremental policy;
20     Remove the last two tokens from the active beam;
21 end
```

In Algorithm 1, the overgeneration problem is addressed by stopping unreliable beams (see Line 9). The unreliable beam is defined as a beam ending with <eos> token or having a score lower or equal to any other unreliable beam detected so far. This means, that we stop any beam that has a score lower than any beam ending with <eos> token. Since there might be a hypothesis that would always score lower than some hypothesis ending with the <eos> token, the algorithm allows generating a hypothesis with a score lower than the unreliable score if it was seen during the decoding of previous blocks.

Finally, the algorithm removes two instead of one token in the current beam (see Line 20). Removing the last two tokens mitigates the issue of low-quality translation toward the end of the context.[1]

2.2 Rethinking Online Policies for Attention-based ST Models

While the improved incremental blockwise beam search improves the performance, it still requires a strong online policy such as hold-n or local agreement (Liu et al., 2020). A common property of these online policies is that they require multiple re-generations of the output translation. For example, the local agreement policy must generate each token at least twice to show it to the user, as each token must be independently generated by two consecutive contexts to be considered stable. Depending on the model architecture, the generation might be the most expensive operation. Additionally, the sequence-to-sequence models tend to suffer from exposure bias (i.e., the model is not exposed to its own errors during the training) (Ranzato et al., 2015; Wiseman and Rush, 2016). The exposure bias then causes a lower translation quality, and sometimes leads to hallucinations (i.e., generation of coherent output not present in the source) (Lee et al., 2018; Müller et al., 2019; Dong et al., 2020). Finally, attentional encoder-decoder models are suspected to suffer from label bias (Hannun, 2020).

A good candidate to address these problems is CTC (Graves et al., 2006). For each input frame, CTC predicts either a blank token (i.e., no output) or one output token independently from its previous predictions, which better matches the streaming translation and reduces the risk of hallucinations. Because the CTC's predictions for each frame are conditionally independent, CTC does not suffer from the label bias problem (Hannun, 2020). Although, the direct use of CTC in either machine or speech translation is possible, yet, its quality lags behind autoregressive attentional modeling (Libovický and Helcl, 2018; Chuang et al., 2021).

[1] Initial experiments showed that removing more than two tokens leads to higher latency without any quality improvement.

Another way, how to utilize the CTC is joint decoding (Watanabe et al., 2017; Deng et al., 2022). In the joint decoding setup, the model has two decoders: the non-autoregressive CTC (usually a single linear layer after the encoder) and the attentional autoregressive decoder. The joint decoding is typically guided by the attentional decoder, while the CTC output is used for re-scoring. Since the CTC predicts hard alignment, the rescoring is not straightforward. To this end, Watanabe et al. (2017) proposed to use the CTC prefix probability (Graves, 2008) defined as a cumulative probability of all label sequences that have the current hypothesis h as their prefix:

$$p_{\text{ctc}}(h, ...) = \sum_{\nu \in \mathcal{V}^+} p_{\text{ctc}}(h \oplus \nu | X), \tag{1}$$

where $\mathcal{V}$ is output vocabulary (including the <eos> symbol), $\oplus$ is string concatenation, and X is the input speech. To calculate this probability effectively, Watanabe et al. (2017) introduce variables $\gamma_t^{(b)}(h)$ and $\gamma_t^{(n)}(h)$ that represent forward probabilities of h at time t, where the superscript denotes whether the CTC paths end with a blank or non-blank CTC symbol. If the hypothesis h is a complete hypothesis (i.e., ends with the <eos> token), then the CTC probability of $h = g \oplus \texttt{<eos>}$ is:

$$p_{\text{ctc}}(h|X) = \gamma_T^{(b)}(g) + \gamma_T^{(n)}(g), \tag{2}$$

where T is the final time stamp.

If $h = g \oplus c$ is not final, i.e., $c \neq \texttt{<eos>}$, then the probability is:

$$p_{\text{ctc}}(h|X) = \sum_{t=1}^{T} \Phi_t(g) \cdot p(z_t = c|X), \tag{3}$$

where

$$\Phi_t(g) = \gamma_{t-1}^{(b)}(g) + \begin{cases} 0 & \text{last}(g) = c \\ \gamma_{t-1}^{(n)}(g) & otherwise. \end{cases}$$

2.3 CTC Online Policy

Based on the the definition of $p_{\text{ctc}}(h|X)$ in Equations (2) and (3), we can define the odds of g being at the end of context T:

$$\text{Odds}_{\text{end}}(g) = \frac{p_{\text{ctc}}(g \oplus \texttt{<eos>}|X)}{\sum_{c \in \mathcal{V}/\{\texttt{<eos>}\}} p_{\text{ctc}}(g \oplus c|X)}. \tag{4}$$

The disadvantage of this definition is that $p_{\text{ctc}}(\ldots|X)$ must be computed for every vocabulary entry separately and one evaluation costs $\mathcal{O}(T)$, i.e., $\mathcal{O}(|\mathcal{V}| \cdot T)$ in total. Contemporary ST systems use vocabularies in orders of thousands items making this definition prohibitively expensive. Since the CTC is used together with the label-synchronous decoder, we can approximate the denominator with a single vocabulary entry c_{att} predicted by the attentional decoder p_{att}:

$$\text{Odds}_{\text{end}}(g) \approx \frac{p_{\text{ctc}}(g \oplus \texttt{<eos>}|X)}{p_{\text{ctc}}(g \oplus c_{\text{att}}|X)}, \tag{5}$$

where $c_{\text{att}} = \text{argmax}_{c \in \mathcal{V}/\{\texttt{<eos>}\}} p_{\text{att}}(g \oplus c|X)$. Now the evaluation of $\text{Odds}_{\text{end}}(g)$ is $\mathcal{O}(T)$. If we consider that the baseline model already uses CTC rescoring, then evaluating $\text{Odds}_{\text{end}}(g)$ amounts to a constant number of extra operations to evaluate $p_{\text{ctc}}(g \oplus \texttt{<eos>}|X)$.

Finally, to control the latency of the online decoding, we compare the logarithm of $\text{Odds}_{\text{end}}(g)$ with a tunable constant C_{end}. If $\log \text{Odds}_{\text{end}}(g) > C_{\text{end}}$, we stop the beam search and discard the last token from g. We found values of C_{end} between -2 and 2 to work well across all models and language pairs.

3 Experiments and Results

3.1 Models

Our offline multilingual ST models are based on attentional encoder-decoder architecture. Specifically, the encoder is based on WavLM (Chen et al., 2022), and the decoder is based on multilingual BART (Lewis et al., 2019) or mBART for short. The model is implemented in the NMTGMinor library.[2] For details on the offline model see KIT submission to IWSLT 2023 Multilingual track (Liu et al., 2023).

The small simultaneous speech translation models for English-to-German and English-to-Chinese language pairs follow the blockwise streaming Transformer architecture (Tsunoo et al., 2021) implemented in ESPnet-ST-v2 (Yan et al., 2023). Specifically, the encoder is a blockwise Conformer (Gulati et al., 2020) with a block size of 40 and look-ahead of 16, with 18 layers, and a hidden dimension of 256. The decoder is a 6-layer Transformer decoder (Vaswani et al., 2017). To improve the training speed, we initialize the encoder with

[2] https://github.com/quanpn90/NMTGMinor

weights pretrained on the ASR task. Further, we employ ST CTC (Deng et al., 2022; Yan et al., 2022) after the encoder with weight 0.3 during the training. During the decoding, we use 0.3 for English to German, and 0.4 for English to Chinese. We preprocess the audio with 80-dimensional filter banks. As output vocabulary, we use unigram models (Kudo, 2018) of size 4000 for English to German, and 8000 for English to Chinese.

3.2 Evaluation

In all our experiments with the offline models, we use beam search of size 8 except for the CTC policy experiments where we use greedy search. For experiments with the blockwise models, we use the beam search of 6. For experiments with the improved blockwise beam search, we follow Polák et al. (2023) and remove the repetition detection in the underlying offline models, while we keep the repetition detection on for all experiments with the blockwise models.

For evaluation, we use Simuleval (Ma et al., 2020) toolkit and `tst-COMMON` test set of MuST-C (Cattoni et al., 2021). To estimate translation quality, we report detokenized case-sensitive BLEU (Post, 2018), and for latency, we report average lagging (Ma et al., 2019). To realistically assess the inference speed, we run all our experiments on a computer with Intel i7-10700 CPU and NVIDIA GeForce GTX 1080 with 8 GB graphic memory.

3.3 Incremental Blockwise Beam Search with Controllable Quality-Latency Tradeoff

In Table 1, we compare the performance of the onlinized version of the baseline blockwise beam search (BWBS) with the improved blockwise beam search (IBWBS; Polák et al., 2023). As we can see in the table, the improved beam search achieves higher or equal BLEU scores than the baseline beam search across all language pairs. We can observe the highest improvement in English-to-German (1.1 BLEU), while we see an advantage of 0.1 BLEU for English-to-Japanese. and no improvement in English-to-Chinese.

In Table 1, we also report the real-time factor (RTF), and the computation-aware average lagging (AL_{CA}). Interestingly, we observe a higher computational footprint of the IBWBS compared to the baseline beam search by 13, 28, and 17 % on En→{De, Ja, Zh}, resp., when measured with RTF. This might be due to the fact that we recompute the decoder states after each source increment. Since the IBWBS sometimes waits for more source chunks to output more tokens, the unnecessary decoder state recomputations might increase the computational complexity.

Lang	Decoding	AL↓	AL_{CA}↓	RTF↓	BLEU↑
En-De	BWBS	1922	**3121**	**0.46**	30.6
	IBWBS	1977	3277	0.52	**31.7**
En-Ja	BWBS	1992	**3076**	**0.50**	15.5
	IBWBS	1935	3264	0.64	**15.6**
En-Zh	BWBS	1948	**2855**	**0.41**	**26.5**
	IBWBS	1945	3031	0.48	**26.5**

Table 1: Incremental SST with the original BWBS and IBWBS. Better scores in bold.

3.4 CTC Online Policy

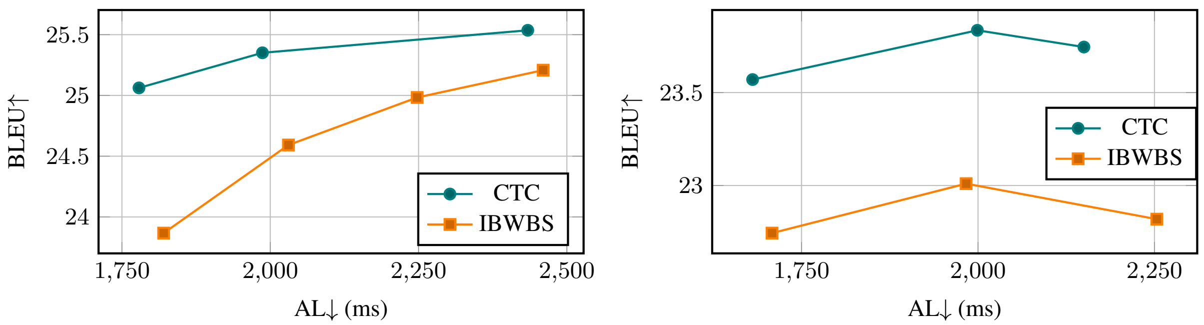
In Figure 1, we compare the improved blockwise beam search (IBWBS) with the proposed CTC policy using the blockwise streaming models. The tradeoff curves for English-to-German (see Figure 1a) and English-to-Chinese (see Figure 1b) show that the proposed CTC policy improves the quality (up to 1.1 BLEU for En→De, and 0.8 BLEU for En→Zh), while it is able to achieve the same latencies.

3.5 CTC Online Policy for Large Offline Models

We were also interested in whether the CTC policy can be applied to large offline models. Unfortunately, due to limited resources, we were not able to train a large offline model with the CTC output. Hence, we decided to utilize the CTC outputs of the online blockwise models and used them to guide the large offline model. Since the models have very different vocabularies,[3] we decided to execute the CTC policy after a whole word is generated by the offline model (rather than after every sub-word token). For the very same reason, we do not use CTC for rescoring.

We report the results in Table 2. Unlike in the blockwise models (see Section 3.4), the CTC policy does not improve the quality in En→De, and has a slightly worse quality (by 0.7 BLEU) in En→Zh. This is most probably due to the delayed CTC-attention synchronization that is not present for the blockwise models (as both decoders there share the

[3]The blockwise models have a vocabulary size of 4000 for En→De and 8000 for En→Zh, and the offline model has 250k.



(a) English to German

(b) English to Chinese

Figure 1: Comparison of the improved blockwise beam search (IBWBS) and the proposed CTC policy using blockwise streaming models.

same vocabulary and the models compute the CTC policy after each token rather than word). However, we still observe a significant reduction in computational latency, namely by 45 and 34 % relative RTF for En→De and En→Zh, respectively.

Lang	Decoding	AL↓	AL_{CA}↓	RTF↓	BLEU↑
En-De	BWBS	1922	3121	0.46	30.6
	IBWBS	1977	3277	0.52	**31.7**
	CTC	1946	**2518**	**0.21**	30.6
En-Zh	BWBS	1948	2855	0.41	**26.5**
	IBWBS	1945	3031	0.48	**26.5**
	CTC	1981	**2515**	**0.28**	25.8

Table 2: Comparison of onlinization of the large offline model using chunking with the local agreement policy (LA-2) and with the proposed CTC policy.

4 Submission

In this section, we summarize our submission to the Simultaneous track at IWSLT 2023. In total, we submit 10 systems for all three language pairs.

4.1 Onlinized Offline Models

Following our last year's submission, we onlinize two large offline models (our models for IWSLT 2022 Offline ST track and IWSLT 2023 Multilingual track). This year, however, we utilize the improved blockwise beam search to yield higher BLEU scores. We submit systems for all language pairs based on the last year's model, and our new model. We summarize the submitted models and their performance in Table 3. As we can observe in Table 3, the 2023 model appears to perform worse. However, we learned during the writing of this paper that there was some overlap between the training and test data for the 2022 model[4], making the BLEU scores for the 2022 model unreliable.

[4](Zhang and Ao, 2022) found an overlap between ST-TED training corpus and tst-COMMON set of MuST-C dataset.

Lang	Model	AL↓	AL_{CA}↓	BLEU↑
En-De	2022	1991	3138	31.8
	2023	1955	3072	31.4
En-Ja	2022	1906	3000	15.5
	2023	1982	3489	15.3
En-Zh	2022	1984	3289	26.8
	2023	1987	3508	26.6

Table 3: Submitted onlinized large offline models.

We also submit the system based on the large model onlinized using the CTC policy. The systems are summarized in Table 4. Unfortunately, we were not aware of the training and test data overlap during the evaluation period, so we decided to use our 2022 model also this year.

Lang	Model	AL↓	AL_{CA}↓	BLEU↑
En-De	2022	1959	2721	31.4
En-Zh	2022	1990	2466	26.3

Table 4: Submitted large offline models onlinized using the proposed CTC policy.

4.2 Blockwise Online Models

Finally, we submit small blockwise models. Their advantage is that they are able to run on a CPU faster than real time (more than 5× faster). We report their performance in Table 5.

Lang	AL↓	AL_{CA}↓	RTF↓	BLEU↑
En-De	1986	2425	0.19	25.4
En-Zh	1999	2386	0.19	23.8

Table 5: Submitted small blockwise models using the proposed CTC online policy.

5 Conclusion and Future Work

In this paper, we present the CUNI-KIT submission to the Simultaneous track at IWSLT 2023. We experimented with the latest decoding methods and proposed a novel CTC online policy. We experimentally showed that the proposed CTC online policy significantly improves the translation quality of the blockwise streaming models. Additionally, the proposed CTC policy significantly lowers the computational footprint of the onlinized large offline models. Unaware of a data overlap issue in 2022, we eventually chose to use our last years' models in the official evaluation also this year.

Acknowledgments

This work has received support from the project "Grant Schemes at CU" (reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935), the grant 19-26934X (NEUREM3) of the Czech Science Foundation, and by Charles University, project GA UK No 244523.

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Yannick Estève, Kevin Duh, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutail Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Ojha Kr., John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. 2023. Findings of the IWSLT 2023 Evaluation Campaign. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Shun-Po Chuang, Yung-Sung Chuang, Chih-Chiang Chang, and Hung-yi Lee. 2021. Investigating the reordering capability in CTC-based non-autoregressive end-to-end speech translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1068–1077, Online. Association for Computational Linguistics.

Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. 2022. Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 1746–1750.

Linhao Dong, Cheng Yi, Jianzong Wang, Shiyu Zhou, Shuang Xu, Xueli Jia, and Bo Xu. 2020. A comparison of label-synchronous and frame-synchronous end-to-end models for speech recognition. *arXiv preprint arXiv:2005.10113*.

Alex Graves. 2008. *Supervised sequence labelling with recurrent neural networks*. Ph.D. thesis, Technical University Munich.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

Awni Hannun. 2020. The label bias problem.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Danni Liu, Ngoc-Quan Pham, Tuan Nam Nguyen, Thai-Binh Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, and Alexander Waibel. 2023. KIT submission to multilingual track at IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Association for Computational Linguistics.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Proc. Interspeech 2020*, pages 3620–3624.

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.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2019. Domain robustness in neural machine translation. *arXiv preprint arXiv:1911.03109*.

Thai-Son Nguyen, Sebastian Stüker, and Alex Waibel. 2020. Super-human performance in online low-latency recognition of conversational speech. *arXiv preprint arXiv:2010.03449*.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Peter Polák, Brian Yan, Shinji Watanabe, Alexander Waibel, and Ondrej Bojar. 2023. Incremental Blockwise Beam Search for Simultaneous Speech Translation with Controllable Quality-Latency Tradeoff. In *Proc. Interspeech 2023*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Emiru Tsunoo, Yosuke Kashiwagi, and Shinji Watanabe. 2021. Streaming transformer asr with blockwise synchronous beam search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 22–29. IEEE.

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.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306.

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W Black, and Shinji Watanabe. 2022. Ctc alignments improve autoregressive translation. *arXiv preprint arXiv:2210.05200*.

Brian Yan, Jiatong Shi, Yun Tang, Hirofumi Inaguma, Yifan Peng, Siddharth Dalmia, Peter Polák, Patrick Fernandes, Dan Berrebbi, Tomoki Hayashi, et al. 2023. Espnet-st-v2: Multipurpose spoken language translation toolkit. *arXiv preprint arXiv:2304.04596*.

Ziqiang Zhang and Junyi Ao. 2022. The YiTrans speech translation system for IWSLT 2022 offline shared task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 158–168, Dublin, Ireland (in-person and online). Association for Computational Linguistics.