NAIST English-to-Japanese Simultaneous Translation System for IWSLT 2021 Simultaneous Text-to-text Task

Ryo Fukuda, Yui Oka, Yausumasa Kano, Yuki Yano, Yuka Ko, Hirotaka Tokuyama, Kosuke Doi, Sakriani Sakti, Katsuhito Sudoh, Satoshi Nakamura
Nara Institute of Science and Technology, Nara, Japan
fukuda.ryo.fo3@is.naist.jp

Abstract

This paper describes NAIST's system for the English-to-Japanese Simultaneous Text-to-text Translation Task in IWSLT 2021 Evaluation Campaign. Our primary submission is based on *wait-k* neural machine translation with sequence-level knowledge distillation to encourage literal translation.

1 Introduction

Automatic simultaneous translation is an attractive research field that aims to translate an input before observing its end for real-time translation similar to human simultaneous interpretation. Starting from early attempts using rule-based machine translation (Matsubara and Inagaki, 1997; Ryu et al., 2006) and statistical methods using statistical machine translation (Bangalore et al., 2012; Fujita et al., 2013), recent studies successfully applied neural machine translation (NMT) into this task (Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019).

The simultaneous translation shared task in the IWSLT evaluation campaign started on 2020 with English-to-German (Ansari et al., 2020) speech-to-text and text-to-text tasks, and a new language pair of English-to-Japanese has been included on 2021 only in text-to-text task. English-to-Japanese is much more challenging than English-to-German due to the large language difference in addition to data scarsity.

We developed an automatic text-to-text simultaneous translation system for this shared task. We applied some extensions to a standard *wait-k* NMT in the training time: sequence-level knowledge distillation and target-side chunk shuffling. However, these techniques showed mixed results in different latency regimes on the IWSLT21 development set, so we configured the system differently for each latency regime. This paper describes the details of the system and the results on the development sets. We also describe our another attempt to include incremental constitutent label prediction that was not included in the primary system.

2 Simultaneous Neural Machine Translation with *wait-k*

Let $X = x_1, x_2, \ldots, x_{|X|}$ be an input sequence in a source language and $Y = y_1, y_2, \ldots, y_{|Y|}$ be an output sequence in a target language. Here, the input can be speech or text, but we assume the input is text because this paper discusses the text-to-text task. The task of simultaneous translation is to translate X to Y incrementally. In other words, each output prediction of Y is made upon partial input observations of X. Suppose an output prefix subsequence $Y_1^j = y_1, y_2, ..., y_j$ has already been predicted from prefix observations of the input $X_1^i = x_1, x_2, ..., x_i$. When we predict the next output subsequence $Y_{j+1}^{j'} = y_{j+1}, ..., y_{j'}$ after further partial observations $X_{i+1}^{i'} = x_{i+1}, ..., x_{i'}$, the prediction is made based on the following formula:

$$Y_{j+1}^{j'} = \underset{\hat{Y}}{\operatorname{argmax}}\, P(\hat{Y}|X_1^i, X_{i+1}^{i'}, Y_1^j) \qquad (1)$$

where $\hat{Y}$ is a possible prediction of the subsequence. In a usual *consecutive* machine translation, we can use the whole input sequence X anytime in the prediction of Y. The limitation of available input information is a key challenge of simultaneous translation.

Wait-k delays the decoding process in k input tokens (Ma et al., 2019). The *wait-k* model translates a token sequence of the source language X into that of the target language Y as follows.

$$\begin{aligned} H_i &= \mathit{Encoder}(x_1, \ldots, x_{i+k-1}), \\ \hat{y}_i &= \mathit{Decoder}(H_i, \hat{y}_1, \ldots, \hat{y}_{i-1}). \end{aligned} \qquad (2)$$

The decoder has to predict an output token based on the attention over an observed portion of the input

tokens. k is a hyperparameter for the fixed delay in this model; setting k larger causes longer delays, while smaller k would result in worse output predictions due to the poor context information.

3 Knowledge Distillation

Knowledge Distillation (KD) (Hinton et al., 2015) is a method that uses the distilled knowledge learned by a stronger teacher model in the learning of a weaker student model. When teacher distribution is $q(y|x;\theta_T)$, we minimize the cross-entropy with the teacher's probability distribution instead of reference data, as follows:

$$\mathcal{L}_{\mathcal{KD}}(\theta;\theta_{\mathcal{T}}) = -\sum_{k=1}^{|\mathcal{V}|} q(y=k|x;\theta_T) \times \log p(y=k|x;\theta) \quad (3)$$

where θ_T parameterizes the teacher distribution.

Sequence-level Knowledge Distillation (SKD), which gives the student model the output of the teacher model as knowledge, propagates a wide range of knowledge to the student model and trains it to mimic its knowledge (Kim and Rush, 2016). The teacher distribution $q(Y|X)$ is approximated by its mode $q(Y|X) \approx \mathbb{1}\{Y = \underset{X \in T}{\operatorname{argmax}}\, q(Y|X)\}$, and the loss objectives as follows:

$$\mathcal{L}_{\mathcal{SKD}} = -\mathbb{E}_{x\sim data} \sum_{Y\in T} q(Y|X) \log p(Y|X)$$
$$\approx -\mathbb{E}_{X\sim data, \widehat{Y}=\underset{Y\in T}{\operatorname{argmax}}\, q(Y|X)} [\log p(y=\widehat{Y}|X)] \quad (4)$$

where $p(Y|X)$ is the sequence-level distribution, and $Y \in T$ is the space of possible target sentences. SKD can be implemented simply by training the student model using $(X, \widehat{Y})$, where $\widehat{Y}$ is derived from the teacher model outputs for the source language portion of the training corpus.

We use SKD for reduction of colloquial expressions in the spoken language corpus. Such colloquial expressions are highly dependent on languages and difficult to translate by NMT, which usually generates literal translations. Here, we firstly train a teacher, Transformer-based *offline* NMT model using the training corpus and use it to obtain pseudo-reference translations in the target language. Then, we train a student, Transformer-based *simultaneous* NMT model using the pseudo-parallel corpus with the original source language sentences and the corresponding translation results by the teacher model. The pseudo-references should consist of more literal and NMT-friendly translations, therefore the training of the student model becomes easier than that using the original parallel corpus. Since we have to train simultaneous translation using less context information than an offline translation model, the SKD would be helpful. This is motivated by the recent success of non-autoregressive NMT using SKD (Gu et al., 2018).

4 Target-side chunk shuffling

Chunk shuffling is a kind of data augmentation that reorders Japanese chunks (called *bunsetsu*). Our motivation for this one is to encourage monotonic IMT utilizing a characteristic of Japanese as an agglutinative language, in which the order of *bunsetsu* chunks is not so strict. When we have a target language sequence $T = t_1, \ldots, t_{|T|}$ in the training set, we apply greedy left-to-right chunking to it; T is divided as a chunk sequence $\bar{T} = \mathcal{C}_1, \ldots, \mathcal{C}_Q$, in which each chunk consists of k (i.e., delay hyperparameter in *wait-k*) tokens $\mathcal{C}_q = t_{q_1}, \ldots, t_{q_k}$. Note that the last chunk $\mathcal{C}_Q$ may be shorter than k according to the length of T. Then, we choose to shuffle or keep the chunks in $\bar{T}$ with a probability p_r, defined as a hyperparameter. We tried only the random shuffling with the fixed chunk size of k in this time; More linguistically-motivated chunk reordering would be worth trying as future work.

5 Primary system

5.1 Implementation

Our system implementation was based on the official baseline[1] using fairseq (Ott et al., 2019) and SimulEval (Ma et al., 2020).

5.2 Setup

Data All of the models were based on Transformer, trained using 17.9 million English-Japanese parallel sentences from WMT20 news task and fine-tuned using 223 thousand parallel sentences from IWSLT 2017. During fine-tuning, we examined the effectiveness of knowledge distillation and chunk shuffling with several hyperparameter settings and reported the results by the models that resulted in the higher BLEU on IWSLT 2021 development set. The text was preprocessed by Byte Pair Encoding (BPE) (Sennrich et al., 2016)

[1] https://github.com/pytorch/fairseq/blob/master/examples/simultaneous_translation/docs/enja-waitk.md

System	BLEU	AL
offline	16.8	-
Baseline		
wait-10	11.8	7.27
wait-20	14.69	11.47
wait-30^{high}	15.57	13.7
Proposed		
wait-10 + CShuf^{low}	13.77	7.29
wait-10 + SKD	13.5	7.28
wait-20 + SKD^{medium}	15.22	11.48
wait-30 + SKD	15.21	13.71

Table 1: In-house results of our systems on IWSLT 2021 En-Ja development set. Superscripts ^{low}, ^{medium} and ^{high} represent the systems submitted for low-, medium-, and high-latency regimes, respectively.

for subword segmentation. The vocabulary was shared over English and Japanese, and its size was 16,000.

Model The hyperparameters of the model almost followed the Transformer Base settings (Vaswani et al., 2017). The encoder and decoder were composed of 6 layers. We set the word embedding dimensions, hidden state dimensions, feed-forward dimensions to 512, 512, and 2,048, respectively. We performed the sub-layer's dropout with a probability of 0.1. The number of attention heads was eight for both the encoder and decoder. The model was optimized using Adam with an initial learning rate of 0.0007, $\beta_1 = 0.9$, and $\beta_2 = 0.98$, following Vaswani et al. (2017).

Evaluation To evaluate the performance, we calculated BLEU and Average lagging (AL) (Ma et al., 2019) with SimulEval on IWSLT 2021 development set.

5.3 Results on the development set

Table 1 shows the excerpt of system results for the full-sentence topline (offline), *wait-k* baselines (wait-k), and our extensions: SKD (+ SKD) and chunk shuffling (+ CShuf).

We tried some different latency hyperparameter values $k = \{10, 12, 14, \ldots, 32\}$ for comparison. Figure 1 plots our BLEU-AL results for *wait-k* and *wait-k*+SKD. It shows that the use of SKD gave some improvements in low-latency settings with $k = \{10, 12, 14\}$, however, the results with larger k were mixed. These results support our assumption on the difficulty of the translation into colloquial expressions discussed in Section 3.

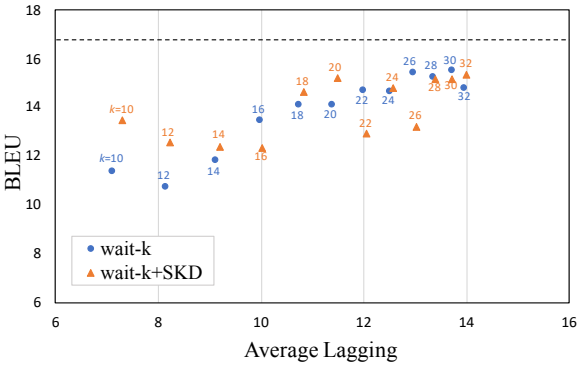

Figure 1: Translation quality against latency for *wait-k* and SKD-based *wait-k* on IWSLT 2021 En-Ja development set. The broken line shows the score of the offline model.

System	p_r	BLEU	len_{hyp}	len_{ref}
Baseline	0	11.80	34,376	27,891
+ CShuf	0.01	10.57	38,257	27,891
	0.02	**13.77**	29,369	27,891
	0.03	9.87	42,296	27,891

Table 2: Target-side chunk shuffling result in $p_r = \{0, 0.01, 0.02, 0.03\}$

We also tried chunk shuffling with different hyperparameter values[2] $p_r = \{0, 0.01, 0.02, 0.03\}$. Table 2 shows the result using the target-side chunk shuffling. Here, the chunk shuffling results are only shown for *wait-10*. The use of larger latency hyperparameter k did not show remarkable differences from the baseline. Chunk shuffling with $p_r = 0.02$ resulted in the best BLEU and outperformed the baseline, but the other values $0.01, 0.03$ did not work. These differences should be due to the output length shown in len_{hyp} column in Table 2; the output length became much shorter than the baseline using the chunk shuffling with $p_r = 0.02$. In contrast, $p_r = 0.01$ and $p_r = 0.03$ increased the output length.

Table 3 shows translation examples by the baseline and chunk-shuffling ($p_r = 0.02$). Here, the baseline translation results do not have end-of-sentence expressions like です (*desu*), ます (*masu*), ですよね (*desuyone*). These differences were not straightforward with the chunk shuffling, but a certain value of $p_r = 0.02$ worked in our experiment.

The results above suggest that the target-side

[2]Higher values of p_r resulted in much worse results and are not included in this paper.

En-input	I see other companies that say, “I’ll win the next innovation cycle, whatever it takes.”
Baseline	他 の 会社 が 「 次 の イノベーション サイクル に <unk> 」 と 言う の は どんな もの で あれ
CShuf	他 の 会社 が 「 次 の イノベーション サイクル に 勝てる 」 と 言う の を 見 ます
Ja-ref	私 の 経験 で も 沢山 の 企業 が 同じ ように 「 何 が なん でも 次 の イノベーション サイクル を 制覇 する 」 と 言い 続け て ます
En-input	She’s a musical instrument maker, and she does a lot of wood carving for a living.
Baseline	彼女 は 楽器 の 製造 者 で 木 彫 り を し て 生き て いる 間 に
CShuf	彼女 は 楽器 の 製作 者 で 木 彫 り を し て い ます
Ja-ref	彼女 は 楽器 の 制作 技師 です 木 を 削る こと で 生計 を 立て て い ます
En-input	Humans are very good at considering what might go wrong if we try something new, say, ask for a raise.
Baseline	人間 は 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が うまく いけ ば 何 が 起き て も 何 が 起き て も 何 が 起き て も 何 が 起き て も 何 が 起き て も 何 が 起き て も 何 が 起き て
CShuf	人間 は 何 が 間違っ て いる の か を 考える の が 得意 です 新しい こと を 試し て み て も いい です よ ね
Ja-ref	昇給 を 求める という よう な 何 か 新しい こと を 試みよ う という とき 人 は どう まずい こと に なり 得る か 考える こと に 長け て い ます

Table 3: Translation examples by *wait-k* baseline and *wait-k* with chunk shuffling ($p_r = 0.02$).

System	BLEU	AL
wait-10 + CShuf^{low}	14.41	7.21
wait-20 + SKD^{medium}	16.20	11.54
wait-30^{high}	16.19	13.83

Table 4: Official results of our submissions on IWSLT 2021 En-Ja test set.

chunk shuffling may work as a perturbation, and we need further investigation.

5.4 Official results on the test set

Table 4 shows BLEU and AL results on the test set. The system with the medium latency regime (wait-20 + SKD) worked relatively well; it achived a comparable BLEU result with wait-30. However, the results were worse than those of the other teams by around two points in BLEU in all the latency regimes.

6 Another attempt: Incremental Next Constituent Label Prediction

We tried another technique described below in the shared task, but it was not included in our primary submission because it did not outperform the baseline. Here, we also describe this for further investigation in future.

For simultaneous machine translation, deciding how long to wait for input before translation is important. Predicting what kind of phrase comes next is a part of useful information in determining the timing. In this study, we tried incremental Next Constituent Label Prediction (NCLP).

In SMT-based simultaneous translation, Oda et al. (2015) proposed a method to predict unseen syntactic constituents to determine when to start

train	**dev**	**eval**
2,762,408	27,903	21,941

Table 5: Number of NCLP instances.

translation for partially-observed input, using a multi-label classifier based on linear SVMs (Fan et al., 2008). Motivated by this study, we used a neural network-based classifier using BERT (Devlin et al., 2019) for NCLP. The problem of NCLP is defined as the label prediction of a syntactic constituent coming next to a given word subsequence in the pre-order tree traversal. In this work, we used 1-lookahead prediction, so the problem was relaxed into the prediction of a label of a syntactic constituent given its preceding words and the first word composing it. A predicted constituent label was inserted at the corresponding position in the input word sequence, immediately after its preceding word. That doubled the length of input sequences. For subword-based NMT, we applied BPE only onto words in the input sequences and put dummy labels after subwords other than end-of-word ones, to order the input in an alternating way.

We used Huggingface transformers (Wolf et al., 2020) for our implementation of NCLP with `bert-base-uncased`. We used Penn Treebank 3 (Marcus et al., 1993) for the NCLP training and development sets, and NAIST-NTT TED Talk Treebank (Neubig et al., 2014) for the NCLP evaluation set. Table 5 shows the number of training, development, and evaluation instances extracted from the datasets. Note that we can extract many instances from a single parse tree.

Table 6 shows the results of the 5 most frequent labels in the NCLP training data. NP and VP are

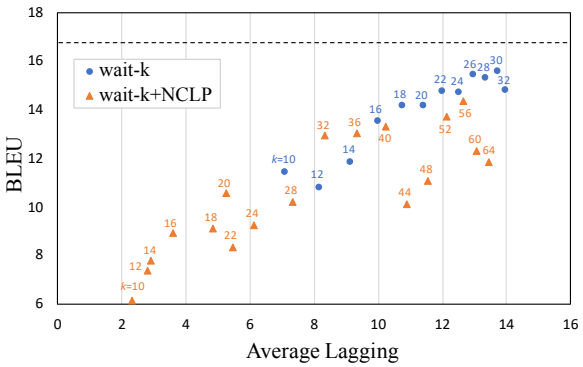

Figure 2: Translation quality against latency for *wait-k* and NCLP-based *wait-k* on IWSLT21 En-Ja dev set. The broken line shows the score of the offline model.

Label	Precision	Recall	F1
NP	0.90	0.94	0.92
VP	0.89	0.97	0.93
NN	0.95	0.97	0.96
,	0.98	1.00	0.99
PP	0.85	0.93	0.89

Table 6: NCLP results on the evaluation set.

important clues of the sentence structure, and their F1 scores were over 90% on the NCLP evaluation data.

However, the results by *wait-k* using NCLP results as its input did not outperform the baseline *wait-k*, as shown in Figure 2. We can observe NCLP-based *wait-k* gave smaller ALs with the same latency hyperparameter k. One possible problem of current NCLP-based *wait-k* is that the length of an input length is doubled by the additional constitutent labels. Since we ran wait-k-based simultaneous NMT for such an augmented input sequence, the decoder using NCLP-augmented input has roughly half of the information compared to the decoder using original input if we use the same k. This forces the decoder to perform very aggressive anticipation with limited information from an input prefix.

Table 7 shows the translation input and output examples of baseline and NCLP. Input sentences include constituents labels. The first example shows that NCLP could translate "publication" before a verb "work" following the Japanese sentence order. Second example shows NCLP output is constructed naturally in terms of grammar, while the baseline has repetitive and unnatural phrases. We observed NCLP sentences are tend to be shorter and more natural than baseline like these examples. However, many sentences are not informative and missing details compared to the baseline. We'll investigate a more effective way to use NCLP in our future work.

7 Conclusion

In this paper, we described our English-to-Japanese text-to-text simultaneous translation system. We extended the baseline *wait-k* with the knowledge distillation to encourage literal translation and target-side chunk shuffling to relax the output order in Japanese. They achieved some improvements on IWSLT 2021 development set in certain latency regimes.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number JP17H06101.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 437–445, Montréal, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference*

En-input	I won't work with you until your publication, or your organization, is more inclusive of all kinds of difference."
En-input (with label)	I VP won NP 't NNP work PP with NP you SBAR until NP your NN public@@ @@@ ation , , CC or NP your NN organization , , VP is ADJP more JJ in@@ @@@ clusive PP of NP all NNS kinds PP of NP difference . . : "
Baseline	出版までは一緒に働かないし組織もすべての種類の違いを包括している」
NCLP	私はあなたと仕事をしません出版されるまでは
Ja-ref	「あなたの出版物や組織が多様性を受け入れるまではご一緒に仕事はできませんと言うこともできます
En-input	Those of us who are underrepresented and invited to participate in such projects, can also decline to be included until more of us are invited through the glass ceiling, and we are tokens no more.
En-input (with label)	Those PP of NP us SBAR who SQ are VP under@@ @@@ represented CC and VP invited PP to VP participate PP in NP such NNS projects , , VP can ADVP also VP decline PP to VP be VP included PP until NP more PP of NP us VP are VP invited PRT through NP the NN glass NN ceiling , , CC and S we VP are NP tok@@ @@@ ens ADVP no RBR more . .
Baseline	私たちはこのようなプロジェクトに参加している人たちは私たちがこのようなプロジェクトに参加している人たちはガラスの天井を通して招待されるまではその人たちはその人たちの中に含まれることを拒否することができますそして私たちはもうトークンを持っていません
NCLP	私たちの代表であり、招待されている人たちは、このようなプロジェクトに参加することもできます。
Ja-ref	過@@小評価されている私たちのような者が招待されたプロジェクトへの参加を断@@ることもできます更に多くの者がガラスの天井をぬ@@け名ばかりの女性参@@画ではなくなるまでは

Table 7: Translation examples by *wait-k* baseline and *wait-k* with NCLP.

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, XiangRui Wang, and Chih-Jen Lin. 2008. *LIBLINEAR: A library for large linear classification*. The Journal of Machine Learning Research.

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Simple, Lexicalized Choice of Translation Timing for Simultaneous Speech Translation. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH-2013)*, pages 3487–3491.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-Autoregressive Neural Machine Translation. In *6th International Conference on Learning Representations (ICLR 2018)*.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

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

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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Shigeki Matsubara and Yasuyoshi Inagaki. 1997. Incremental Transfer in English-Japanese Machine Translation. *IEICE Transactions on Information and Systems*, E80-D(11):1122–1130.

Graham Neubig, Katsuhito Sudoh, Yusuke Oda, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2014. The NAIST-NTT TED talk treebank. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Syntax-based simultaneous translation through prediction of unseen syntactic constituents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 198–207, Beijing, China. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Koichiro Ryu, Shigeki Matsubara, and Yasuyoshi Inagaki. 2006. Simultaneous English-Japanese spoken language translation based on incremental dependency parsing and transfer. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 683–690, Sydney, Australia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.