End-to-End Simultaneous Translation System for the IWSLT2020 using Modality Agnostic Meta-Learning

Houjeung Han, Mohd Abbas Zaidi, Sathish Indurthi,
Nikhil Kumar Lakumarapu, Beomseok Lee, Sangha Kim
Next AI Solution Lab, Samsung Research, Seoul, South Korea
{h.j.han, abbas.zaidi, s.indurthi, n07.kumar, bsgunn.lee, sangha01.kim}@samsung.com

Abstract

In this paper, we describe end-to-end simultaneous speech-to-text and text-to-text translation systems submitted to IWSLT2020 online translation challenge. The systems are built by adding wait-k and meta-learning approaches to the Transformer architecture. The systems are evaluated on different latency regimes. The simultaneous text-to-text translation achieved a BLEU score of 26.38 compared to the competition baseline score of 14.17 on the low latency regime (Average latency $\leq$ 3). The simultaneous speech-to-text system improves the BLEU score by 7.7 points over the competition baseline for the low latency regime (Average Latency $\leq$ 1000).

1 Introduction

Simultaneous Neural Machine Translation (SNMT) addresses the problem of live interpretation in machine translation. In a traditional neural machine translation model, the encoder first reads the entire source sequence, and then the decoder generates the translated target sequence. On the other hand, a simultaneous neural machine translation model alternates between reading the source sequence and writing the target sequence using either a fixed or an adaptive policy. This would allow the model to avoid intolerable delay in live or streaming translation scenarios.

In this work, we build a simultaneous translation system for text-to-text(t2t) and speech-to-text(s2t) problems based on Transformer wait-k model (Ma et al., 2019a). We adopt the meta-learning approach presented in (Indurthi et al., 2020) to deal with the data scarcity issue in the speech-to-text translation task. The system architecture and data processing techniques are designed for the IWSLT 2020 online translation task (Ansari et al., 2020). However, these techniques can be applied to current and future SNMT models as well. We conduct several experiments on both text-to-text and speech-to-text problems to evaluate the proposed system. Our experimental results reveal that the proposed system achieves significant performance gains over the provided competition baselines on both the translation tasks.

2 Simultaneous Translation

2.1 Base Model

The machine translation task involves converting an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $x_i \in \mathbb{R}^{d_x}$ in the source language to the output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_k)$, $y_t \in \mathbb{R}^{d_y}$ in the target language. In the simultaneous translation task, the model produces the output in an online fashion as the input is read. Hence, while producing an output y_t, the complete input sequence might not have been processed .

Our model derives from the transformer wait-k model proposed in (Ma et al., 2019a). Similar to (Ma et al., 2019b), the encoder consists of unidirectional transformer blocks unlike bi-directional transformer blocks used in the original transformer. The decoder starts producing the translation after having read the first k input units from the source sequence. It learns to anticipate the information which might be missing due to word order differences between the input and target sequences. The model also supports training and testing under different latency regimes, i.e., different k.

In the machine translation task, the input and output can have different lengths. This difference is highly prominent for language pairs such as English-Chinese. The average source to target ratio for each language pair(r) is defined as $r = \overline{|\mathbf{y}|/|\mathbf{x}|}$ and the catch-up frequency is defined as $c = r - 1$. For the speech-to-text translation task, $|\mathbf{x}|$ is set to the length of the transcript of input waveform and we define stride, s, which represents the number

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 62–68

https://doi.org/10.18653/v1/P17

of frames in the source waveform to be consumed in order to produce each target text token. Usually, s is set to 1 for the text-to-text translation task. The wait-k model adjusts the reading speed of the decoder according to this ratio r and stride s. Hence, the final decoding policy of the model can be defined by the the following equation:

$$g_{wait-k,\,c,s}(t) = \min\{(k + t - 1 - \lfloor ct \rfloor) * s, |x|\},$$

where $g(t)$ is the number of input units processed in order to produce y_t.

2.2 Meta Learning

Recently, (Indurthi et al., 2020) proposed a Modality Agnostic Meta-Learning (MAML, (Finn et al., 2017)) approach to address the data scarcity issue in the speech-to-text translation task. We adopt this approach to train our simultaneous translation task. Here, we briefly describe the MAML approach used for training, for more details, please refer to (Indurthi et al., 2020).

The MAML approach involves two steps: (1) Meta-Learning Phase, (2) Fine-tuning Phase. In the meta-learning phase, we use a set of related high resource tasks as source tasks to train the model. In this phase, the model captures the general learning aspects of the tasks involved. In the fine-tuning phase, we initialize the model from the parameters learned during the meta-learning phase and train further to learn the specific target task.

Meta-Learning Phase: The set of source tasks involved in the meta-learning phase are denoted by T. For each step in this phase, we first uniformly sample one source task $\tau \in T$ and then sample two batches(D_τ and $D_\tau^{'}$) of training examples. The D_τ is used to train the model to learn the task specific distribution, and this step is called meta-train step. In each meta-train step, we create auxiliary parameters (θ_τ^a) initialized from the original model parameters (θ^m). We update the auxiliary parameters during this step while keeping the original parameters of the model intact. The auxiliary parameters (θ^a) are updated using gradient-descent steps, which is given by,

$$\theta_\tau^a = \theta^m - \alpha \nabla_{\theta^m} \ell(D_\tau; \theta^m). \quad (1)$$

After the meta-train step, the auxiliary parameters (θ^a) are evaluated on $D_\tau^{'}$. This step is called meta-test and the gradients computed during this step are used to update the original model parameters(θ^m).

$$\theta_\tau^m = \theta^m - \beta \nabla_{\theta^a} \ell(D_\tau^{'}; \theta^a).` \quad (2)$$

Exposing the meta-learned parameters((θ^m) to the vast data of the source tasks T during this phase makes them suitable to act as a good initialization point for the future related target tasks.

Fine-tuning Phase: During the fine-tuning phase, the model is initialized from the meta-learned parameters (θ^m) and trained on a specific target task. In this phase, the model training is carried out like a usual neural network training without involving the auxiliary parameters.

2.3 Training

We train our systems with and without using the meta-learning approach described in Section 2.2. In the meta-learning approach, we first pre-train the model on the source tasks and further fine-tune on the target task, which is represented as 'wMT'. We also train another model directly on the given target task without using the meta-learning approach, represented as 'woMT'. The meta-learning training approach helps the low resource tasks to utilize the training examples from the high resource source tasks.

The source tasks used for simultaneous speech-to-text translation are Automatic Speech Recognition (ASR), Machine Translation (MT), and Speech Translation (ST) tasks. Unlike (Indurthi et al., 2020), we also added the ST task as a source task, and this improved the performance of our system further. Even though the simultaneous text-to-text translation task has sufficient training data, we apply the meta-learning training approach to learn possible language representations across different language pairs. We use English-German and French-German language pairs as the source tasks in the meta-training for the text-to-text translation task.

The text sequences are represented as word-piece tokens, and the speech signals are represented as Log Mel 80-dimensional features. Usually, the speech sequences are a few times longer than the text sequences, therefore, we use an additional layer to compress the speech signal and exploit structural locality. The compression layer consists of 3 Convolution layers with stride 2, both on the time and frequency domain of the speech sequence. The compressed speech sequence is passed to the encoder layer for further processing. To facilitate

Pair \ Dataset	Must-C	OpenSubtitles	WMT19	All
EnDe	229k	22.5m	38m	61m
FrDe	-	-	9.8m	9.8m

Table 1: Dataset Statistics for T2T

Dataset \ Wait-k	4	7	8	15	Offline
All	25.50	28.31	28.80	29.70	-
All + BT	25.06	28.22	28.71	-	-
All_reduced	26.04	28.57	29.07	30.27	31.20

Table 2: Comparing Datasets for T2T

the training on multiple language pairs and tasks, we create a universal vocabulary ((Gu et al., 2018a)) for both text-to-text and speech-to-text translation systems. The universal vocabularies are created based on the source and target tasks.

For each simultaneous task, we train the system on a dataset D of parallel sequences to maximize the the log likelihood:

$$\ell(D;\theta) = \frac{1}{|D|}\sum_{i=1}^{|D|} \log p\left(\boldsymbol{y}^i|\boldsymbol{x}^i;\theta\right), \qquad (3)$$

where θ denotes the parameters of the model. We train the systems for three different latency regimes based on the competition requirements.

3 Experiments

3.1 Datasets

3.1.1 Simultaneous Text-to-Text Translation

For the text-to-text translation task, we use the MuST-C, IWSLT 2020, OpenSubtitles2018, and WMT19 (presented as 'All' in the Table 2) for training. We evaluate our system on the MuST-C Dev set. Our parallel corpus of WMT19 consists of Europarl v9, ParaCrawl v3, Common Crawl, News Commentary v14, Wiki Titles v1 and Document-split Rapid for the German-English language pair. We also use the WMT19 German-French language pair as one of the source tasks during the meta-learning training phase. The statistics of the data we use for text-to-text translation are provided in the Table 1. We also use monolingual data from the News crawl corpus for data augmentation using back-translation technique. About 20M English sentences are translated by the En-De translation model, which was trained on the WMT19 corpus (presented as 'All + BT' in the Table 2). Due to the presence of noise in the OpenSubtitles2018 and ParaCrawl, we use only 10M randomly sampled examples from these corpora (presented as 'All_reduced' in the Table 2).

Task	Dataset	Hours	Sent. #
MT	Open Subtitles	-	22.5m
MT	WMT 19	-	4.6m
ASR	LibriSpeech	982	233k
ASR	IWSLT 19 ST	272	145k
ASR	MuST-C	400	229k
ASR	TED LIUM 3	452	28.6k
ST	Europarl-ST	89	97.9k
ST	IWSLT ST 19	272	726k
ST	MuST-C	400	918k
ST	TED-LIUM 3	452	537k

Table 3: Dataset Statistics for S2T

3.1.2 Simultaneous Speech-to-Text Translation

The speech-to-text translation models are trained on examples collected from the Must-C, IWSLT 2020, Europarl-ST, and TED-LIUM3 datasets. The statistics for the same are provided in the Table 3. The models are evaluated using the MuST-C Dev set. Due to the limited availability of training examples for the ST task, we increase the number of training examples by using data augmentation techniques. For data augmentation on the text side, we use English-to-German NMT model to generate synthetic German sequences from the English sequences. We use two NMT models and top-K beam results to generate multiple synthetic German sequences. These NMT models are based on the Transformer architecture and trained on the WMT19 dataset with different hyper-parameter settings. For speech sequence, we use the Sox library to generate the speech signal using different values of speed, echo, and tempo parameters similar to (Potapczyk et al., 2019). The parameter values are uniformly sampled using these ranges for each parameter: tempo $\in (0.85, 1.3)$, speed $\in (0.95, 1.05)$, echo_delay $\in (20, 200)$, and echo_decay $\in (0.05, 0.2)$. We increase the size of the IWSLT2020 ST dataset to five times of the original size by augmenting 4X data – four text sequences using the NMT models and four speech signals using the Sox parameter ranges. For the Europarl-ST, we augment 2X examples to triple the size. The TED-LIUM3 dataset does not contain speech-to-text translation examples originally, hence, we create 2X synthetic speech-to-text trans-

Train-k^t	4		7		8		26		27		28	
Decode-k	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL
k^t	26.04$^{\oplus}$	2.84	28.57*	5.15	29.07	5.85	30.78	14.59	30.16	14.93	30.48	15.17
$k^t - 1$	24.32	2.06	28.24	4.47	29.07*	5.15	30.73	14.27	30.13	14.60	30.48	14.89
$k^t - 3$	17.98	0.32	26.38$^{\oplus}$	2.93	27.75	3.71	30.69	13.58	30.21	13.96	30.49	14.27

Table 4: Varying k during Testing for T2T: k^t denotes k used for training

Latency regimes		Low			Medium			High		
Methods	Dataset	BLEU	AL	k	BLEU	AL	k	BLEU	AL	k
Fairseq	Must-C	14.17	2.91	4	17.28	5.88	8	19.53	12.37	20
woMT	All_reduced	26.04	2.84	4	29.07	5.85	8	30.78	14.59	26
wMT	All_reduced	25.31	2.83	4	28.75	5.76	8	30.08	14.57	26

Table 5: Comparing Training Strategies for T2T

lations using speech-to-text transcripts. Finally, for the MuST-C datasest, we use synthetic speech to increase the dataset size to 4X. Overall, we created the synthetic training data of size roughly equal to two times the original data using data augmentation techniques described above.

3.2 Implementation Details

For the text-to-text translation, we use `base` parameter settings from the Transformer model (Vaswani et al., 2017), except that we use unidirectional encoder. Each model is trained for 500k steps with a batch size of 4096 tokens. The source tasks used in the meta-training phase are English-German and French-German language pairs. The stride s is set to 1.

For the speech-to-text translation, the number of encoder and decoder layers are 8 and 6, respectively. The compression layer consists of three Convolutional layers. Each model is trained for 300k meta steps and fine-tuned for another 300k with a text batch size of 4096 tokens and a speech batch size of 1.5M frames. The models are trained using the multi-step Adam optimizer (Saunders et al., 2018) with the gradients accumulated over 32 steps.

Our code is based on the Tensor2Tensor framework (Vaswani et al., 2018), and we use 4*NVIDIA P40 GPUs for all the experiments. We use the server-client API based evaluation code provided by the organizers of the IWSLT2020 online translation challenge. This evaluation API gives several metrics to measure the translation quality and latency, such as BLEU, METEOR, TER, Differentiable Average Lagging(DAL), Average Lagging(AL) and Average Proportion (AP). In this paper, we report the BLEU scores along with the AL. We also report the numbers from the baselines provided by the organizing committee in the Table 5 and 7. All the results are reported on the MuST-C Dev set, unless stated otherwise. The emission rate r of German-English is set to 1.0. Moreover, we use the same parameter value for k and s during training and testing, unless stated otherwise.

4 Results

4.1 Simultaneous Text-to-Text Translation

We train our models on different dataset sizes which are created by using back translation and sampling techniques and compare the performance across these datasets. The BLEU scores with various wait-k values for models trained on different dataset sizes have been reported in the Table 2. As we can see in the Table 2, the augmented dataset ('All + BT') performs poorly compared to the model trained on the original dataset. On the contrary, the reduced dataset gives best performance among all these datasets. All these models are trained using the *woMT* training strategy.

Motivated by (Ma et al., 2019a), we decode the target text using smaller k values than the k value used during the training. As one can observe (by comparing the marked cells) in the Table 4, the result obtained from a model upon decoding using $k = 7$, when trained using $k = 8$ is better than the model which is both trained and decoded using $k = 7$. A similar trend is also observed for $k = 4$. Also, as train-k or decode-k increases, usually the BLEU and the AL also increases. However, this trend is limited to the low or medium latency regimes, since the models with larger k are less sensitive to k value and the performance degrades as k reaches towards the input sequence length. For

Train-k^t/s^t	3/300		4/350		3/400		4/800		5/800	
Decode-k/s	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL
k^t/s^t	12.85	1136.11	14.59	1875.04	15.89	1940.67	17.95	3967.49	17.42	4318.34
$k^t - 1/s^t$	12.24	897.79	13.88	1539.73	14.9	1653.62	17.79	3582.93	17.43	4002.83
$k^t/s^t - 100$	7.16	45.48	10.77	715.92	12.79	1084.11	17.81	3679.68	17.14	4027.91

Table 6: Varying k/s during Testing for S2T: k^t/s^t denote k/s used during training

Latency regimes		Low				Medium				High			
Methods	Dataset	BLEU	AL	k	s	BLEU	AL	k	s	BLEU	AL	k	s
Fairseq	Must-C	4.5	792.28	1	320	9.3	1828.28	2	400	11.49	3664.19	2	800
woMT	iwslt20_aug	6.70	1061.90	3	300	9.11	1882.17	3	400	12.59	4020.97	4	800
wMT	iwslt20_aug	12.85	1136.11	3	300	15.89	1940.67	3	400	17.95	3967.49	4	800

Table 7: Comparing the Training Strategies for S2T

example, the model trained with $k = 26$ has the highest BLEU score among the models trained on k values ranging from 26 to 28. All the models reported in the Table 4 use the All_reduced dataset and the *woMT* training approach.

We compare the *wMT* and *woMT* training strategies on three latency regimes. The results have been tabulated in the Table 5. Unlike speech translation, we did not witness any improvement in the text-translation from the meta-learning approach. In the Table 5, models trained using the *woMT* strategy achieved a better results than the *wMT* strategy. A possible reason for this might be that the English-German text translation problem is not suffering from data scarcity. Moreover, the number or diversity of source tasks used for meta-learning training is limited compared to the speech-to-text translation source tasks. We also observe that English-German and French-German corpus have an overlap of over 70% German words limiting the variability of the source tasks, which hampers the model from learning any meta-aspects of the text translation tasks during the meta-learning phase. This might be the reason behind meta-training being less effective for online text-to-text task.

4.2 Simultaneous Speech-to-Text Translation

Similar to the online text-to-text task, we vary the latency parameters while decoding the simultaneous speech model as well. We vary both k and strides(s), and report the BLEU in the Table 4. We can see from the Table 6 that as k and s increase, the BLEU score increases while AL decreases. Unlike the text-to-text translation, decoding with decreased k and s does not result in any BLEU score improvement. For instance, as seen in the Table 6, the result of model trained with $5/800$(where $k = 5$ and stride $s = 800$) and decoded with $4/800$ shows lower performance than that of the model both trained and decoded using $4/800$. Also, a similar trend can be observed between the models trained with $3/400$ and $3/300$. As we can see in the last two columns, the BLEU score decreases as k increases from 4 to 5 for $s = 800$. This is similar to what we observed in the text case as well, increasing k in the high latency regime leads to a drop in the performance. All the models reported in the Table 6 are trained using the augmented datasets and the *wMT* training approach.

Finally, we explore the effectiveness of the meta-learning approach for the online speech-to-text translation task for the three latency regimes. In the Table 7, we can easily see that there is a significant BLEU score gap between models trained using the *wMT* and *woMT* training strategy. The results show that our meta-learning approach improves the performance of the models in all the latency regimes. Compared to the online text-to-text translation task, the meta-learning approach in the online speech-to-text task exploits many sub-problems with a variety of source tasks such as ASR, MT and ST. Also, the speech-to-text task suffers severely from the data-scarcity issue as compared to the text-to-text task, and using the meta-learning approach helps overcome this issue.

5 Related Work

Simultaneous Translation: The earlier works in simultaneous translation such as (Cho and Esipova, 2016; Gu et al., 2016; Press and Smith, 2018; Dalvi et al., 2018) lack the ability to anticipate the words with missing source context. The wait-k model introduced by (Ma et al., 2019a) brought in many improvements by introducing a simultaneous trans-

lation module which can be easily integrated into most of the sequence to sequence models. Arivazhagan et al. (2019) introduced MILk which is capable of learning an adaptive schedule by using hierarchical attention; hence it performs better on the latency quality trade-off. Wait-k and MILk are both capable of anticipating words and achieving specified latency requirements by varying the hyper-parameters.

Speech to Text Translation: Most of the existing systems tackle the problem of simultaneous speech to text translation using a cascaded pipeline of online ASR and MT systems.

Previous works such as (Niehues et al., 2016);(Ari et al., 2020) propose re-translation strategies for simultaneous speech translation, but their use case is limited to settings where output revision is allowed.

Although there has been some work towards making an end-to-end offline speech translation modules, the paucity of training datasets remains a bottleneck. The work done by (Ma et al., 2019a) cannot simply be extended to the domain of simultaneous speech translation as we discussed earlier. Similar to our model (Gu et al., 2016) also uses a pre-trained model for the simultaneous translation task. However, they use a full-sentence model during pre-training, unlike ours. Our proposed model alleviates these issues, both our pre-training and fine-tuning training phases are done in an online fashion, hence avoiding any train-inference discrepancies. Our model has a controllable latency which can be specified by k.

Meta Learning: Meta-Learning approaches have been particularly useful with low resource problems since they inherently learn to adapt to a new problem with less training examples. Andrychowicz et al. (2016); Ha et al. (2016) focuses more on the meta policy while MAML system proposed by (Finn et al., 2017) focuses more on finding a good initialization point for the target tasks. The work done by (Indurthi et al., 2020) and (Gu et al., 2018b) employ the MAML algorithm for low resource settings in offline speech-to-text and text-to-text translation task. In this work, we adopt these strategies to the online translation tasks.

6 Conclusion

In this work, we develop an end-to-end simultaneous translation system for both text-to-text and speech-to-text tasks by using the wait-k method and the meta-learning approach. We evaluate the proposed system with different data settings and latency regimes. We explore the effectiveness of the meta-learning approach for the online translation tasks. The meta-learning approach proves to be essential in settings where the training data is scarce. Compared to the baseline provided in the competition, both online text-to-text and speech-to-text models achieved significant BLEU score improvements.

References

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej 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, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

Naveen Ari, Colin Andrew Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020. Re-translation strategies for long form, simultaneous, spoken language translation.

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. *arXiv preprint arXiv:1906.05218*.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018a. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2016. Learning to translate in real-time with neural machine translation. *arXiv preprint arXiv:1610.00388*.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018b. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

S. Indurthi, H. Han, N. K. Lakumarapu, B. Lee, I. Chung, S. Kim, and C. Kim. 2020. End-end speech-to-text translation with modality agnostic meta-learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7904–7908.

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. 2019a. 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, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2019b. Monotonic multihead attention. *arXiv preprint arXiv:1909.12406*.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic transcription for low-latency speech translation. In *Interspeech*, pages 2513–2517.

Tomasz Potapczyk, Pawel Przybysz, Marcin Chochowski, and Artur Szumaczuk. 2019. Samsung's system for the iwslt 2019 end-to-end speech translation task. Zenodo.

Ofir Press and Noah A Smith. 2018. You may not need attention. *arXiv preprint arXiv:1810.13409*.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. Multi-representation ensembles and delayed SGD updates improve syntax-based NMT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 319–325, Melbourne, Australia. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

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.