AN EMPIRICAL STUDY OF END-TO-END SIMULTANEOUS SPEECH TRANSLATION DECODING STRATEGIES

Ha Nguyen, Yannick Estève, Laurent Besacier

AN EMPIRICAL STUDY OF END-TO-END SIMULTANEOUS SPEECH TRANSLATION DECODING STRATEGIES

Ha Nguyen[1,2]*, Yannick Estève*[2]*, Laurent Besacier*[1,3]

[1]LIG - Université Grenoble Alpes, France
[2]LIA - Avignon Université, France
[3]Naver Labs Europe, France

ABSTRACT

This paper proposes a decoding strategy for end-to-end simultaneous speech translation. We leverage end-to-end models trained in offline mode and conduct an empirical study for two language pairs (English-to-German and English-to-Portuguese). We also investigate different output token granularities including characters and Byte Pair Encoding (BPE) units. The results show that the proposed decoding approach allows to control BLEU/Average Lagging trade-off along different latency regimes. Our best decoding settings achieve comparable results with a strong cascade model evaluated on the simultaneous translation track of IWSLT 2020 shared task.

Index Terms— Simultaneous speech translation, end-to-end models, low-latency decoding.

1. INTRODUCTION

Simultaneous (online) machine translation consists in generating an output hypothesis before the entire input sequence is available [1, 2]. To deal with this low latency constraint, several strategies were proposed for neural machine translation with input text [3, 4, 5]. Only a few works investigated low latency neural speech translation [6, 7, 8]. At IWSLT 2020 workshop, a *simultaneous translation track* was proposed and attempted to stimulate research on this challenging task [9]. In 2020, the best system of this track [7] was made up with a cascade of an ASR system trained using Kaldi [10] and an online MT system with *wait-k* policies [11, 3]. Only one end-to-end simultaneous system was proposed [8] but its performance was not at the level of the cascaded model.

In this paper, we propose a simple but efficient decoding approach that allows to leverage any pre-trained end-to-end speech translation (offline) model for simultaneous speech translation. We conduct an empirical study of the decoding parameters for different language pairs and for different output token granularities (characters or BPEs). Our contributions are the following:

- We adapt the algorithm from [8] but introduce the possibility to write several output tokens at a time.
- We show that it allows to control AL/BLEU trade-off along different latency regimes with a pre-trained end-to-end AST model that does not need to be re-trained in simultaneous mode.
- We conduct an empirical evaluation of our decoding strategies for 2 different language pairs (English-to-Portuguese (EN-PT) and English-to-German (EN-DE)) and with different output granularities (characters or BPEs).
- We show that our best system, evaluated on the same IWSLT 2020 simultaneous shared task dataset, is competitive with the winning (cascade) system of the IWSLT 2020 shared task [7].

2. END-TO-END SIMULTANEOUS DECODING STRATEGIES FOR AST

2.1. Architecture of the end-to-end model

As mentioned in [12], we use an attention-based encoder-decoder architecture, whose encoder is a stack of 2 VGG-like [13] CNN blocks, and then 5 layers of 1024-dimensional BLSTM [14]. Each VGG block consists of two 2D-convolution layers, followed by a 2D-maxpooling layer. These two VGG blocks transform the shape of input speech features from $(T \times D)$ to $(T/4 \times D/4)$, with T being the length of the input sequence (number of frames), and D being the features' dimension. We use Bahdanau's attention mechanism [15] throughout all the experiments presented in this paper. The decoder is a stack of two 1024-dimensional LSTM layers.

2.2. Simultaneous decoding strategies

Our end-to-end simultaneous decoding strategy is inspired by *wait-k* decoding strategy introduced for text NMT in [3]. It is also built on [8] who proposed the only end-to-end approach at the simultaneous translation track of IWSLT 2020. However, their models were based on Transformer and a meta-learning approach was needed to obtain decent results in low latency regimes. Transformer-based *wait-k* models also required re-training in low latency mode in [16]. In this work,

we hypothesize that simpler LSTM encoders might be more robust to limited source context when no re-training is performed in online mode. Consequently, our (LSTM-based) end-to-end models trained in offline mode are re-used without any adaptation nor re-training in this work.

For online decoding, we propose a deterministic strategy[1] with the following parameters (see figure 1):

- k (*wait* parameter) denotes the number of acoustic frames (at the beginning of the input speech features sequence) read before writing the first output token (k=100 or 200 frames in our experiments which is equivalent to 1s or 2s),
- s (*stride* parameter) represents the number of acoustic frames in the input speech features sequence to be consumed in order to produce each new target token (s=10 or 20 in our experiments which is 0.1s or 0.2s),
- N (*write* parameter) represents the number of output tokens written at each decoding step (N=1, 2 or 3 in our experiments, output tokens can be characters or BPEs).

In details, let $\mathbf{X} = (x_1, x_2, ..., x_{|\mathbf{X}|})$ be the source audio sequence, and $\mathbf{Y} = (y_1, y_2, ..., y_{|\mathbf{Y}|})$ be the target hypothesis. Let us also assume that $g(t)$ denotes the number of source frames consumed by the encoder at each decoding step t, and $q(t)$ is the number of target tokens generated up to step t. In this work, $g(t) = min\{k + (t-1) * s, |\mathbf{X}|\}$, and $q(t) = q(t-1) + w_t$, with $0 \leq w_t \leq N$ being the number of target tokens emitted at step t, and $q(0) = 0$. At each decoding step t, the model encodes $g(t)$ source frames in order to decode at maximum N target tokens.

Since our pre-trained offline AST models are based on Bidirectional Long Short Term Memory (BLSTM) networks, one has to re-encode from beginning of the source sequence every time new frames are read:

$$h^t = encode(\mathbf{X}^t) \tag{1}$$

with $\mathbf{X}^t = x_{\leq g(t)} = (x_1, x_2, ..., x_{g(t)})$ being input buffer at step t.

The decoder then takes the encoder's hidden states sequence h^t, and the cached previous hidden state $z_{q(t-1)}$ to compute w_t hidden states, and predict w_t corresponding target tokens. With $j \in [q(t-1)+1, q(t)]$, we have:

$$z_j = decode(h^t, z_{j-1}, y_{j-1}) \tag{2}$$

$$y_j = predict(z_j) \tag{3}$$

We update the output buffer $\mathbf{Y}^t$ by simply appending $(y_{q(t-1)+1}, ..., y_{q(t)})$ to $\mathbf{Y}^{t-1}$:

$$\mathbf{Y}^t = \mathbf{Y}^{t-1} + (y_{q(t-1)+1}, ..., y_{q(t)}) \tag{4}$$

This process ends when the end of sequence token </s> is predicted, or the length of the output buffer $|\mathbf{Y}|$ exceeds a threshold.[2] In case the decoder generates the </s> token before the whole source sequence $\mathbf{X}$ is read, we only append the tokens preceding </s>, then break the loop and read more source frames.

[1] We leave dynamic decoding with adaptive read/write policy for future work

Table 1: (BLEU / AL) scores of the EN-DE char model evaluated on MuST-C tst-HE and MuST-C tst-COMMON. Sorted by AL of tst-HE in increasing order. AL is in $milliseconds$.

No.	k	s	N	tst-HE	tst-COMMON
1	100	10	3	3.01 / 743	4.42 / 800
2	100	10	2	4.26 / 1049	6.89 / 1135
3	100	20	3	5.49 / 1353	8.61 / 1441
4	200	10	3	7.07 / 1521	10.37 / 1552
5	200	10	2	8.77 / 1836	12.79 / 1931
6	100	20	2	8.77 / 2062	12.68 / 2097
7	100	10	1	9.46 / 2146	12.85 / 2157
8	200	20	3	10.38 / 2223	14.6 / 2286
9	200	20	2	13.8 / 2934	16.59 / 2840
10	200	10	1	14.11 / 2973	16.83 / 2880
11	100	20	1	14.64 / 3487	15.79 / 3086
12	200	20	1	17.15 / 4066	17.94 / 3610
13	offline			20.54 / 7005	21.38 / 5782

2.3. Evaluation of simultaneous speech translation

In order to measure the latency, we use an adaptive version [17] of Average Lagging (AL) introduced by [3]. The original AL metric measures the average rate (in *milliseconds* in our case) that the MT system lags behind an ideal wait-0 translator:

$$AL_g(\mathbf{X}, \mathbf{Y}) = \frac{1}{\tau_g(|\mathbf{X}|)} \sum_{t=1}^{\tau_g(|\mathbf{X}|)} g(t) - \frac{t-1}{\gamma} \tag{5}$$

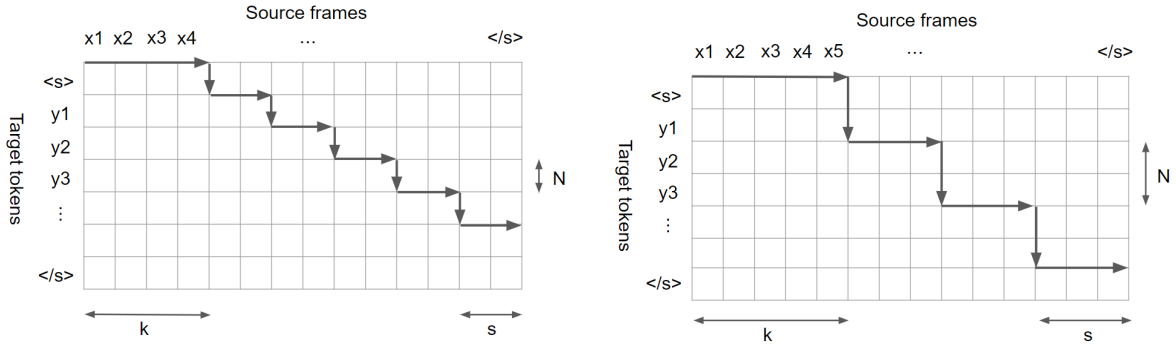
The "cut-off" step $\tau_g(|\mathbf{X}|) = min\{t|g(t) = |\mathbf{X}|\}$ is defined as the decoding step when the policy first finishes reading all source frames, and the target-to-source length ratio $\gamma = |\mathbf{Y}|/|\mathbf{X}|$ is a scale factor accounting for the source and target having different sequence lengths. In our case, since the decoder can emit more than one target token at step t, equation (5) becomes:

$$AL_g(\mathbf{X}, \mathbf{Y}) = \frac{1}{\tau_g(|\mathbf{X}|)} \sum_{t=1}^{\tau_g(|\mathbf{X}|)} [g(t) - \frac{t-1}{\gamma}] * w_t \tag{6}$$

However, as observed by [17], the original metric often generates negative values when applied to speech-to-text translation. Therefore, they propose an adaptive version which computes γ as: $\gamma = |\boldsymbol{Y}^*|/|\mathbf{X}|$, with $\boldsymbol{Y}^*$ being the reference sentence. Tables and curves presented in this paper are computed using the adaptive AL version unless stated otherwise.

In this work, AL is computed at word-level. This means that with models that generate target token units smaller than

[2] in this work, we set a $max\_length\_ratio = \frac{max\_output\_sequence\_length}{encoder\_hidden\_state\_sequence\_length} = 1.0$ for EN-DE experiments, and 1.6 for EN-PT experiments.



(a) N=1 (b) N=2

Fig. 1: Illustration of our simultaneous decoding strategy: k source frames are read before 1st decoding step where N tokens are generated; then s additional frames are read before the next decoding step, etc. Different values of (k, s, N) are displayed in (a) and (b).

word, for example character or BPEs, we wait until a complete word is merged, before committing the corresponding delay to the AL computing module.[3]

Table 2: (BLEU / AL) scores of the EN-PT char model evaluated on MuST-C tst-COMMON. Sorted by AL in increasing order. AL is in $milliseconds$.

No.	k	s	N	tst-COMMON
1	100	10	3	4.67 / 611
2	100	10	2	8.38 / 907
3	100	20	3	11.43 / 1237
4	200	10	3	11.61 / 1402
5	200	10	2	15.97 / 1748
6	100	20	2	16.67 / 1948
7	100	10	1	16.95 / 1976
8	200	20	3	17.94 / 2106
9	200	20	2	20.69 / 2697
10	200	10	1	20.98 / 2735
11	100	20	1	20.64 / 2910
12	200	20	1	22.67 / 3505
13	offline			25.07 / 5986

3. EXPERIMENTAL SETUP

EN-PT pair. As for the EN-PT offline models, we re-use our best offline character-based (char) model, and two BPE-based (BPE400, and BPE2K) models, which were trained for IWSLT 2019 [12] shared task. As mentioned in [12], these models were trained on a merged version of MuST-C EN-PT [18], and How2 corpus [19]. Being trained on this combination of about 674.4 hours of training data, the char model scored 26.91 BLEU, while the BPE400 and BPE2K model scored 24.73, and 23.11 BLEU respectively on MuST-C tst-COMMON set, in beam search mode ($beam\_size = 10$).[4]

EN-DE pair. The offline char EN-DE was trained for our participation to IWSLT 2020 [7]. MuST-C EN-DE, Europarl EN-DE [20], and How2 synthetic (i.e. English speech - German synthetic translation from the original transcription) were merged together in order to train this model. It scored 23.55 and 22.35 BLEU on MuST-C tst-COMMON, and MuST-C tst-HE, in beam search mode ($beam\_size = 10$), respectively.

All of these models were trained using normalized (mean and variance normalization) 83-dimensional Mel filter-bank and pitch features. We consistently use speed perturbation with factors of 0.9, 1.0, and 1.1 as one of the data augmentation methods in all of our experiments. Besides, *SpecAugment* [21] is used to train our EN-DE char model as well. Further details can be found in [12, 7].

4. SIMULTANEOUS DECODING RESULTS

4.1. Impact of decoding parameters

The experimental results (BLEU for different AL) are given in Table 1 for EN-DE and in Table 2 for EN-PT for the character-based models. For a fair comparison with the online mode, the offline models were re-decoded in greedy decoding mode. This gives the results on the last rows of both the tables. We observe that: (a) the 3 parameters of our simultaneous decoding strategy allow to move over the whole range of AL (from very low latency regimes <1s to higher AL values between 2s and 3s), (b) for a latency of 2s, the strategy reaches decent BLEU scores (BLEU=14.60 for EN-DE and BLEU=17.94 for EN-PT on tst-COMMON), (c) writing two characters at each decoding step (N=2) with bigger stride (s=20) seems to be slightly better in term of AL, yet slightly worse in term of

[3] We re-use the latency class from https://github.com/facebookresearch/SimulEval for AL computing.

[4] Our performance in simultaneous mode will be, in contrast, given with greedy decoding mode.

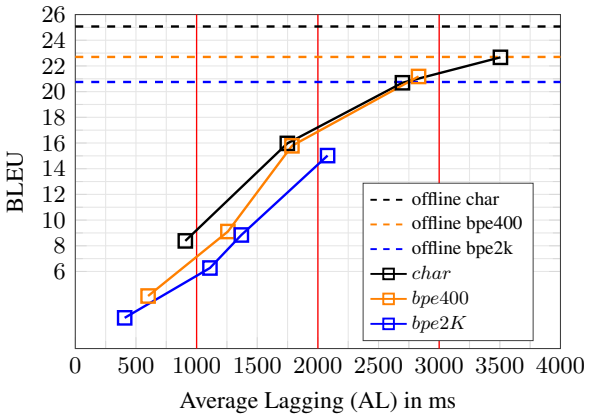


Fig. 2: Character-based vs bpe-based models on English-to-Portuguese translation, evaluated on MuST-C tst-COMMON.

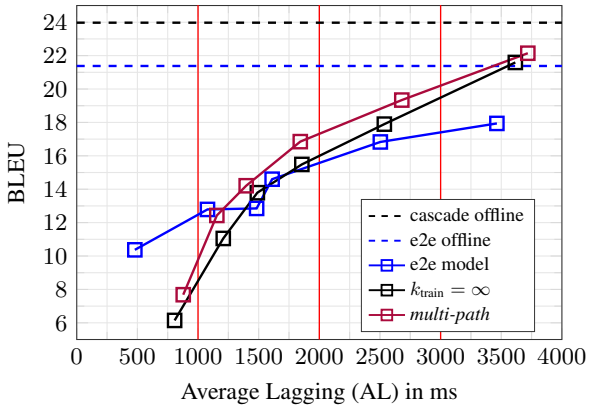


Fig. 3: Comparison of our proposed character-based EN-DE end-to-end model (*e2e model*) with that of the (winning) ON-TRAC cascaded models with (*multi-path*) or without ($k_{train} = \infty$) re-training for simultaneous mode. Original AL metric [3] is used to compute the latency illustrated in this curve.

BLEU scores than writing one character at a time (N=1) with smaller stride (s=10) (compare lines 6-7 and 9-10 of Table 1 and 2 for instance).

4.2. Impact of target granularity

In order to investigate the impact of different target token types, we decode our pre-trained offline BPE2K and BPE400 models trained for IWSLT 2019 [12], using the presented simultaneous decoding strategy. We alternate between different (k, s, N) triplets, with $k = [100, 200]$, $s = [10, 20]$, and $N = [1, 2]$. We then sort the results by the increasing order of AL of each model, and pick the (k, s, N) combinations that give the best BLEU-AL trade-offs. The results are shown in figure 2, whose data points correspond to $(k, s, N) = (100, 10, 2), (200, 10, 2), (200, 20, 2), (200, 20, 1)$. Results are given as a trend and should be taken with caution since offline char-based and BPE-based models have different performance as well. However, figure shows that in general, for a same (k, s, N) triplet, char model tends to give higher BLEU score, yet bigger AL than the two BPE models. We can observe the same trend when comparing BPE400 with BPE2K model. This could be explained by the fact that since BPEs are bigger token units than characters, when given the same amount of context (number of source frames), and forced to generate approximately the same number of target tokens, BPE models unsurprisingly give worse translation quality. However, it takes less number of source frames for BPE models to make up a word in comparison with char model, consequently resulting in smaller AL for BPE models with similar (k, s, N) settings.

4.3. Comparison to the state-of-the-art

At IWSLT 2020, the winning system for speech-to-text online translation paired an ASR system with an online MT system and decoded following a deterministic algorithm described in [7]. The ASR system was a strong hybrid HMM/DNN system built using the Kaldi speech recognition toolkit [10] (WER=14.2% on MuST-C tst-COMMON dataset in offline mode). The online MT system was a Transformer-based [22] *wait-k* decoder with unidirectional encoder. Instead of optimizing a single decoding path, [7] jointly optimized across multiple wait-k paths. The additional loss terms provide a richer training signal, and yield models that perform well under different lagging constraints.

In Figure 3, we choose the (k, s, N) triplets for the EN-DE char model that give close AL regimes to that of the above cascaded models. It is clear that our proposed decoding strategy (the blue curve) performs reasonably well in comparison with the cascaded models with (*multi-path*) or without ($k_{train} = \infty$) re-training for simultaneous mode (*e2e* is better in low latency regimes and *cascade* is better in higher latency regimes). We recall that our *e2e model* did not require re-training in simultaneous mode, it is thus fair to compare it with the corresponding cascaded model: $k_{train} = \infty$. We note that since [7] use the original AL metric to compute their latency, in this curve, we also use original AL [3] to compute the latency of our end-to-end model.

5. CONCLUSION

We propose in this paper a simple yet efficient simultaneous decoding strategy, which allows pre-trained end-to-end AST offline model to decode in simultaneous mode. Our empirical evaluation, conducted on models trained for 2 different language pairs EN-DE and EN-PT, with either characters or BPEs, shows that this strategy can give decent BLEU-AL trade-offs, and its best settings achieve competitive results in comparison with a strong cascade baseline, without re-training the models in online mode (this will however be investigated in future work).

6. ACKNOWLEDGEMENTS

This work was funded by the French Research Agency (ANR) through the ON-TRAC project under contract number ANR-18-CE23-0021, and was performed using HPC resources from GENCI-IDRIS (Grant 20XX-AD011011365).

7. REFERENCES

[1] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez, "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*, 2012, pp. 437–445.

[2] Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan, "Segmentation strategies for streaming speech translation," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 230–238.

[3] 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, "STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proc. of ACL*, 2019.

[4] Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proc. of ACL*, 2019.

[5] Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu, "Monotonic multihead attention," in *Proc. of ICLR*, 2020.

[6] Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel, "Low-latency neural speech translation," in *Proc. of INTERSPEECH*, 2018.

[7] Maha Elbayad, Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Antoine Caubrière, Benjamin Lecouteux, Yannick Estève, and Laurent Besacier, "ON-TRAC Consortium for End-to-End and Simultaneous Speech Translation Challenge Tasks at IWSLT 2020," in *The International Conference on Spoken Language Translation ACL - 17th IWSLT*, Seattle, WA, United States, July 2020.

[8] Hou Jeung Han, Mohd Abbas Zaidi, Sathish Reddy Indurthi, Nikhil Kumar Lakumarapu, Beomseok Lee, and Sangha Kim, "End-to-end simultaneous translation system for IWSLT2020 using modality agnostic meta-learning," in *Proceedings of the 17th International Conference on Spoken Language Translation*, Online, July 2020, pp. 62–68, Association for Computational Linguistics.

[9] 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, "Findings of the IWSLT 2020 Evaluation Campaign," in *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA, 2020.

[10] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, Georg Stemmer, et al., "The Kaldi speech recognition toolkit," in *In IEEE 2011 workshop*, 2011.

[11] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel, "Incremental decoding and training methods for simultaneous translation in neural machine translation," in *Proc. of NAACL-HLT*, 2018.

[12] 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," in *Proc. of IWSLT*, 2019.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of ICLR*, 2015.

[14] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, 1997.

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. of ICLR*, 2015.

[16] Maha Elbayad, Laurent Besacier, and Jakob Verbeek, "Efficient Wait-k Models for Simultaneous Machine Translation," in *Interspeech 2020*, Shangai (Virtual Conf), China, Oct. 2020.

[17] Changhan Wang Jiatao Gu Juan Pino Xutai Ma, Mohammad Javad Dousti, "Simuleval: An evaluation toolkit for simultaneous translation," in *Proceedings of the EMNLP*, 2020.

[18] Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi, "Must-c: a multilingual speech translation corpus," in *Proc. of NAACL-HLT*, 2019.

[19] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze, "How2: a large-scale dataset for multimodal language understanding," in *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018.

[20] Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan, "Europarl-ST: A multilingual corpus for speech translation of parliamentary debates," in *Proc. of ICASSP*, 2020.

[21] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. of INTERSPEECH*, 2019.

[22] 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. of NeurIPS*, 2017.