Visualization: The Missing Factor in Simultaneous Speech Translation

Sara Papi[1,2], **Matteo Negri**[1], **Marco Turchi**[1]
1. Fondazione Bruno Kessler, Italy
2. University of Trento, Italy
{spapi,negri,turchi}@fbk.eu

Abstract

Simultaneous speech translation (SimulST) is the task in which output generation has to be performed on partial, incremental speech input. In recent years, SimulST has become popular due to the spread of multilingual application scenarios, like international live conferences and streaming lectures, in which on-the-fly speech translation can facilitate users' access to audio-visual content. In this paper, we analyze the characteristics of the SimulST systems developed so far, discussing their strengths and weaknesses. We then concentrate on the evaluation framework required to properly assess systems' effectiveness. To this end, we raise the need for a broader performance analysis, also including the user experience standpoint. We argue that SimulST systems, indeed, should be evaluated not only in terms of quality/latency measures, but also via task-oriented metrics accounting, for instance, for the visualization strategy adopted. In light of this, we highlight which are the goals achieved by the community and what is still missing.

1 Introduction

Simultaneous speech translation (SimulST) is the task in which the translation of a source language speech has to be performed on partial, incremental input. This is a key feature to achieve low latency in scenarios like streaming conferences and lectures, where the text has to be displayed following as much as possible the pace of the speech. SimulST is indeed a complex task in which the difficulties of performing speech recognition from partial inputs are exacerbated by the problem to project meaning across languages. Despite the increasing demand for such a system, the problem is still far from being solved.

So far, research efforts mainly focused on the quality/latency trade-off, i.e. producing high quality outputs in the shortest possible time, balancing the need for a good translation with the necessity of a rapid text generation. Previous studies, however, disregard how the translation is displayed and, consequently, how it is actually perceived by the end users. After a concise survey of the state of the art in the field, in this paper we posit that, from the users' experience standpoint, output visualization is at least as important as having a good translation in a short time. This raises the need for a broader, task-oriented and human-centered analysis of SimulST systems' performance, also accounting for this third crucial factor.

2 Background

As in the case of offline speech translation, the adoption of cascade architectures (Stentiford and Steer, 1988; Waibel et al., 1991) was the first attempt made by the SimulST community to tackle the problem of generating text from partial, incremental input. Cascade systems (Fügen, 2009; Fujita et al., 2013; Niehues et al., 2018; Xiong et al., 2019; Arivazhagan et al., 2020b) involve a pipeline of two components. First, a streaming automatic speech recognition (ASR) module transcribes the input speech into the corresponding text (Wang et al., 2020; Moritz et al., 2020). Then, a simultaneous text-to-text translation module translates the partial transcription into target-language text (Gu et al., 2017; Dalvi et al., 2018; Ma et al., 2019; Arivazhagan et al., 2019). This approach suffers from *error propagation*, a well-known problem even in the offline scenario, where

the transcription errors made by the ASR module are propagated to the MT module, which cannot recover from them as it does not have direct access to the audio. Another strong limitation of cascaded systems is the *extra latency* added by the two-step pipeline, since the MT module has to wait until the streaming ASR output is produced.

To overcome these issues, the direct models initially proposed in B[Pleaseinsertintopreamble]rard et al. (2016; Weiss et al. (2017) represent a valid alternative that is gaining increasing traction (Bentivogli et al., 2021). Direct ST models are composed of an encoder, usually bidirectional, and a decoder. The encoder starts from the audio features extracted from the input signal and computes a hidden representation; the decoder transforms this representation into target language text. Direct modeling becomes crucial in the simultaneous scenario, as it reduces the overall system's latency due to the absence of intermediate symbolic representation steps. Despite the data scarcity issue caused by the limited availability of speech-to-translation corpora, the adoption of direct architectures showed to be promising (Weiss et al., 2017; Ren et al., 2020; Zeng et al., 2021), driving recent efforts towards the development of increasingly powerful and efficient models.

3 Architectural Challenges

This section surveys the direct SimulST models developed so far, highlighting strengths and weaknesses of the current architectures and decision policies – i.e. the strategies used by the system to decide whether to output a partial translation or to wait for more audio information. We discuss ongoing research on architectural improvements of encoder-decoder models, as well as popular approaches like offline training and re-translation. All these works concentrate on reducing systems latency, targeting a better quality/latency trade-off.

Encoding Strategy. Few studies (Elbayad et al., 2020a; Nguyen et al., 2021b) tried to improve the encoder part of simultaneous systems. Elbayad et al. (2020a) and Nguyen et al. (2021b) introduced the use of unidirectional encoders instead of standard bidirectional encoders (i.e. the encoder states are not updated after each read action) to speed up the decoding phase. Nguyen et al. (2021b) also proposed an encoding strategy called *Overlap-and-Compensate*, where the encoder exploits extra frames provided from the past that were discarded during the previous encoding step. The segmentation problem is a crucial aspect in SimulST, where the system needs to split a long audio input into smaller chunks (speech frames) in order to process them. Different segmentation techniques can be adopted to extract this information, starting from the easiest one based on fixed time windows (Ma et al., 2020b) to the dynamic ones based on automatically detected word boundaries (Zeng et al., 2021; Chen et al., 2021). Ma et al. (2020b) also studied the dynamic segmentation based on oracle boundaries but they discovered that, in their scenario, it had worse performance compared to that of the fixed segmentation.

Decoding Strategy. Some efforts have been made to improve the decoding strategy as it strongly correlates to the decision policy of simultaneous systems. Speculative beam search, or SBS, (Zheng et al., 2019c) represents the first successful attempt to use beam search in SimulST. This technique consists in hallucinating several prediction steps in the future in order to make more accurate decisions based on the best "speculative" prediction obtained. Also Zeng et al. (2021) integrate the beam search in the decoding strategy, developing the wait-k-stride-N strategy. In particular, the authors bypass output speculation by directly applying beam search, after waiting for k words, on a word stride of size N (i.e., on N words at a time) instead of one single word as prescribed by the standard wait-k. Nguyen et al. (2021a) analyzed several decoding strategies relying on different output token granularities, such as characters and Byte Pair Encoding (BPE), showing that the latter yields lower latency.

Offline or Online training? An alternative approach to simultaneous training is the offline (or full-sentence) training of the system and its subsequent use as a simultaneous one. Nguyen et al. (2021a) explored this solution with an LSTM-based direct ST system, analyzing the effectiveness of different decoding strategies. Interestingly, the offline approach does not only preserve overall performance despite the switch of modality, it also improves system's ability to generate well-formed sentences. These results are confirmed by Chen et al. (2021), who successfully exploit a direct ST system jointly trained in an offline fashion with an

ASR one.

Another point of view: re-translation. Re-translation (Niehues et al., 2016; Niehues et al., 2018; Arivazhagan et al., 2020a; Arivazhagan et al., 2020b) consists in re-generating the output from scratch (e.g. after a fixed amount of time) for as long as new information is received. This approach ensures high quality (the final output is produced with all the available context) and low latency (partial translations can be generated with fixed, controllable delay). This, however, comes at the cost of strong output instability (the so-called *flickering*, due to continuous updates of the displayed translations) which is not optimal from the user experience standpoint. To this end, some metrics have been developed to measure the instability phenomenon, such as the *Erasure* (Arivazhagan et al., 2020b), which measures the number of tokens that were deleted from the emitted translation to produce the next translation.

Decision Policy. In simultaneous settings, the model has to decide, at each time step, if the available information is enough to produce a partial translation – i.e. to perform a *write* action using the information received until that step (audio chunk/s in case of SimulST or token/s in case of simultaneous MT) – or if it has to wait and perform a *read* action to receive new information from the input. Possible decision policies result in different ways to balance the quality/latency trade-off. On one side, more read actions provide the system with larger context useful to generate translations of higher quality. On the other side, this counterbalances the increased, sometimes unacceptable latency. To address this problem, two types of policy have been proposed so far: fixed and adaptive. While *fixed* decision policies look at the number of ingested tokens (or speech chunks, in the speech scenario), in the *adaptive* ones the decision is taken by also looking at the contextual information extracted from the input.

While little research focused on adaptive policies (Gu et al., 2017; Zheng et al., 2019a; Zheng et al., 2020) due to the hard and time-consuming training (Zheng et al., 2019b; Arivazhagan et al., 2019), the adoption of very easy-to-train fixed policies is the typical choice. Indeed, the most widely used policy is a fixed one, called *wait-k* (Ma et al., 2019). Simple yet effective, it is based on waiting for k source words before starting to generate the target sentence, as shown in Table 1.

source	It	was	a	way	that	parents	...
wait-3	-	-	-	Es	ging	um	eine
wait-5	-	-	-	-	-	Es	ging

Table 1: wait-k policy example with $k = \{3, 5\}$

As the original wait-k implementation is based on textual source data, Ma et al. (2020b) adapted it to the audio domain by waiting for k fixed time frames (audio chunks or speech frames) rather than k words. However, this simplistic approach does not consider various aspects of human speech, such as different speech rates, duration, pauses, and silences. In (Ren et al., 2020), the adaptation was done differently, by including a Connectionist Temporal Classification (CTC)-based (Graves et al., 2006) segmentation module that is able to determine word boundaries. In this case, the wait-k strategy is applied by waiting for k pauses between words that are automatically detected by the segmenter. Similarly, Zeng et al. (2021) employed the CTC-based segmentation method but applying a *wait-k-stride-N* policy to allow re-ranking during the decoding phase. The *wait-k-stride-N* model emits more than one word at a time, slightly increasing the latency, since the output is prompted after the stride is processed. This small increase in latency, however, allows the model to perform beam search on the stride, which has been shown to be effective in improving translation quality (Sutskever et al., 2014). Decoding more than one word at a time is the approach also employed by Nguyen et al. (2021a), who showed that emitting two words increases the quality of the translation without any relevant impact on latency. Another way of applying the wait-k strategy was proposed by Chen et al. (2021), where a streaming ASR system is used to guide the direct ST decoding. They look at the ASR beam to decide how many tokens have been emitted within the partial audio segment, hence having the information to apply the original wait-k policy in a straightforward way. An interesting solution is also the one by Elbayad et al. (2020a), who jointly train a direct model across multiple wait-k paths. Once the sentence has been encoded, they optimize the system by uniformly sampling the k value for the decoding step. Even though they reach good performance by using a single-path training with k=7 and a different k value for testing, the multi-path approach proved to be effective. One of its advan-

tages is that no k value has to be specified for the training, which allows to avoid the training from scratch of several models for different values of k.

Retrospective. All the aspects analyzed in this section highlight several research directions already taken by the simultaneous community, which have to be studied more in depth. Among all, the audio or text segmentation strategy clearly emerges as a fundamental factor of simultaneous systems, and the ambivalent results obtained in several studies point out that this aspect has to be better clarified. Moreover, the presence of extensive literature on the wait-k policy shows that it represents one of the topics of greatest interest to the community, which continues to work on it to further improve its effectiveness as it directly impacts on the systems' performance, especially latency. Unfortunately, all these studies focus on the architecture enhancements and decision policies despite the absence of a unique and clear evaluation framework to perform a correct and complete analysis of the system.

4 Evaluation Challenges

A good simultaneous model should produce a high quality translation with reasonable timing, as waiting too long will negatively affect the user experience. Offline MT and ST communities commonly use the well-established BLEU metric (Papineni et al., 2002; Post, 2018) to measure the quality of the output translation, but a simultaneous system also needs a metric that accounts for the time spent by the system to output the partial translation. Simultaneous MT (SimulMT) is the task in which a real-time translation is produced having a partial source text at disposal. Since SimulMT was the first yet easiest simultaneous scenario studied by the community, a set of metrics was previously introduced for the textual input-output translation part.

Latency Metrics for SimulMT. The first metric, the *Average Proportion* (AP), was proposed by Cho and Esipova (2016) and measures the average proportion of source input read when generating a target prediction, that is the sum of the tokens read when generating the partial target. However, AP is not length-invariant, i.e. the value of the metric depends on the input and output lengths and is not evenly distributed on the [0, 1] interval (Ma et al., 2019), making this metric strongly unreliable.

To overcome all these problems, Ma et al. (2019) introduced the *Average Lagging* (AL) that directly describes the lagging behind the ideal policy, i.e. a policy that produces the output exactly at the same time as the speech source. As a downside, Average Lagging is not differentiable, which is, instead, a useful property, especially if the metric is likely to be added in the system's loss computation. For this reason, Cherry and Foster (2019) proposed the *Differential Average Lagging* (DAL), introducing a minimum delay after each operation.

Another way of measuring the lagging is to compute the alignment difficulty of a source-target pair. Hence, Elbayad et al. (2020b) proposed the *Lagging Difficulty* (LD) metric that exploits the use of the `fast-align` (Dyer et al., 2013) tool to estimate the source and target alignments. Then, they infer the reference decoding path and compute the AL metric. The authors claimed the LD to be a realistic measure of the simultaneous translation as it also evaluates how a translation is easy to align considering the context available when decoding.

Latency Metrics for SimulST. The most popular AP, AL and DAL metrics were successively adapted by the SimulST community to the speech scenario by converting, for instance, the number of words to the sum of the speech segment durations, as per (Ma et al., 2020a). Later, Ma et al. (2020b) raised the issue of using computational unaware metrics and proposed computational aware metrics accounting for the time spent by the model to generate the output. Unfortunately, computing such metrics is not easy at all in absence of a unique and reproducible environment that can be used to evaluate the model's performance. To this end, Ma et al. (2020a) proposed *SimulEval*a tool which computes the metrics by simulating a real-time scenario with a server-client scheme. This toolkit automatically evaluates simultaneous translations (both text and speech) given a customizable agent that can be defined by the user and that will depend on the adopted policy. Despite the progress in the metrics for evaluating quality and latency, no studies have been conducted on the effective correlation with user experience. This represents a missing key point in the current evaluation framework landscape, giving rise to the need for a tool that combines quality and latency metrics with application-oriented metrics (e.g., read-

ing speed), which are strongly correlated to the visualization and, as an ultimate goal, to the user experience.

5 The Missing Factor: Visualization

In the previous section, we introduced the most popular metrics used to evaluate the simultaneous systems' performance. These metrics account for the quality and the latency of the system without capturing the user needs. Although many researchers acknowledge the importance of human evaluation, this current partial view can push the community in the wrong direction, in which all the efforts are focused on the quality/latency factors while the problem experienced by the user is of another kind. Indeed, the third factor that matters and strongly influences the human understanding of a – even very good – translation is the *visualization strategy* adopted. The visualization problem and the need to present the text in a readable fashion for the user was only faced in our previous work (Karakanta et al., 2021). In the paper, we raised the need for a clearer and less distracting visualization of the SimulST system's generated texts by presenting them as subtitles (text segmented in lines preserving coherent information). We proposed different visualization strategies to better assess the online display problem, attempting to simulate a setting where human understanding is at the core of our analysis.

Visualization modalities. The standard *word-for-word* visualization method (Ma et al., 2019), in which the words appear sequentially on the screen as they are generated, could be strongly suboptimal for the human understanding (Romero-Fresco, 2011). Infact, the word-for-word approach has two main problems: *i)* the emission rate of words (some go too fast, some too slow) is irregular and the users waste more time reading the text because their eyes have to make more movements, and *ii)* emission of pieces of text that do not correspond to linguistic units/chunks, requiring more cognitive effort. Moreover, when the maximum length of the subtitle (that depends on the dimensions of the screen) is reached, the subtitle disappears without giving the user enough time to read the last words emitted. As this will negatively impact the user experience, we propose in (Karakanta et al., 2021) to adopt different visualization modes that better accommodate the human reading requirements. We first introduced the *block* visualization mode, for which an entire subtitle is displayed at once (usually one or two lines maximum) as soon as the system has finished generating it. This display mode is the easiest to read for the user because it prevents re-reading phenomena (Rajendran et al., 2013) and unnecessary/excessive eye fixations (Romero-Fresco, 2010), reducing the human effort. However, we discovered that the latency introduced by waiting for an entire subtitle is too high to let this visualization mode be used in many simultaneous scenarios. As a consequence, we proposed the *scrolling lines* visualization mode that displays the subtitles line by line. Every time a new line becomes available, it appears at the bottom of the screen, while the previous (older) line is scrolled to the upper line. In this way, there are always two lines displayed on the screen. To evaluate the performance of the system in the different visualization modes, we also proposed an ad-hoc calculation of the *reading speed* (characters per second or CPS) that correlates with the human judgment of the subtitles (Perego et al., 2010). The reading speed shows how fast a user needs to read in order not to miss any part of the subtitle. The lower the reading speed, the better is the model's output since a fast reading speed increases the cognitive load and leaves less time to look at the image. The scrolling line method offers the best balance between latency and a comfortable reading speed resulting to be the best choice for the simultaneous scenario. On the other hand, this approach requires segmented text (i.e. a text that is divided into subtitles), thus the system needs to be able to simultaneously generate transcripts or translations together with proper subtitle delimiters. However, building a simultaneous subtitling system combines the difficulties of the simultaneous setting with the constraint of having a text formatted in proper subtitles. Since both these research directions are still evolving, a lot of work is required to achieve good results.

The lack of studies on this aspects highlights the shortcomings of the actual SimulST systems, individuating possible improvements that will allow the systems to evolve in a more organic and complete way according to the user needs. Moreover, to completely assess the subtitling scenario, a system has to be able to jointly produce timestamps metadata linked to the word emitted, a task that has not been addressed so far. The need for this kind

of system represents an interesting direction to follow for the simultaneous community. In the light of this, the researcher should also take into account the three quality-latency-visualization factors in their analyses. We are convinced that these are the most promising aspects to work on to build the best SimulST system for the audience and that human evaluation has to have a crucial role in future studies. We also believe that interdisciplinary dialogue with other fields such as cognitive studies, media accessibility and human-computer interaction would be very insightful to evaluate SimulST outputs from communicative perspectives (Fantinuoli and Prandi, 2021).

6 Conclusions and Future Directions

SimulST systems have become increasingly popular in recent years and many efforts have been made to build robust and efficient models. Despite the difficulties introduced by the online framework, these models have rapidly improved, achieving comparable results to the offline systems. However, many research directions have not been explored enough (e.g., the adoption of dynamic or fixed segmentation, the offline or the online training). First among all, the visualization strategy that is adopted to display the output of the simultaneous systems is an important and largely under-analyzed aspect of the simultaneous experience. We posit that the presence of application-oriented metrics (e.g., reading speed), which are strongly related to the visualization and, as an ultimate goal, to the user experience, is the factor that misses in the actual evaluation environment. Indeed, this paper points out that BLEU and Average Lagging are not the only metrics that matter to effectively evaluate a SimulST model, even if they are fundamental to judge a correct and real-timed translation. We hope that this will inspire the community to work on this critical aspect in the future.

Acknowledgement

This work has been carried out as part of the project Smarter Interpreting (`https://kunveno.digital/`) financed by CDTI Neotec funds.

References

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, July. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020a. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online, July. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Isabelle Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020b. Re-translation strategies for long form, simultaneous, spoken language translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923. IEEE.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. Cascade versus direct speech translation: Do the differences still make a difference? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2873–2887, Online, August. Association for Computational Linguistics.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. In *NIPS Workshop on end-to-end learning for speech and audio processing*, Barcelona, Spain, December.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2021. Direct simultaneous speech-to-text translation assisted by synchronized streaming ASR. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4618–4624, Online, August. Association for Computational Linguistics.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation?

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, June. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020a. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. Interspeech 2020*, pages 1461–1465.

Maha Elbayad, Michael Ustaszewski, Emmanuelle Esperança-Rodier, Francis Brunet-Manquat, Jakob Verbeek, and Laurent Besacier. 2020b. Online versus offline NMT quality: An in-depth analysis on English-German and German-English. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5047–5058, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Claudio Fantinuoli and Bianca Prandi. 2021. Towards the evaluation of automatic simultaneous speech translation from a communicative perspective. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 245–254, Bangkok, Thailand (online), August. Association for Computational Linguistics.

C. Fügen. 2009. A system for simultaneous translation of lectures and speeches.

Tomoki Fujita, Graham Neubig, S. Sakti, T. Toda, and Satoshi Nakamura. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *INTERSPEECH*.

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*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

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, April. Association for Computational Linguistics.

Alina Karakanta, Sara Papi, Matteo Negri, and Marco Turchi. 2021. Simultaneous speech translation for live subtitling: from delay to display. In *Proceedings of the 1st Workshop on Automatic Spoken Language Translation in Real-World Settings (ASLTRW)*, pages 35–48, Virtual, August. Association for Machine Translation in the Americas.

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, July. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. 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, October. Association for Computational Linguistics.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587, Suzhou, China, December. Association for Computational Linguistics.

Niko Moritz, Takaaki Hori, and Jonathan Le. 2020. Streaming automatic speech recognition with the transformer model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021a. An empirical study of end-to-end simultaneous speech translation decoding strategies. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7528–7532. IEEE.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021b. Impact of Encoding and Segmentation Strategies on End-to-End Simultaneous Speech Translation. In *Proc. Interspeech 2021*, pages 2371–2375.

J. Niehues, T. Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, S. Stüker, and A. Waibel. 2016. Dynamic transcription for low-latency speech translation. In *INTERSPEECH*.

J. Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and A. Waibel. 2018. Low-latency neural speech translation. In *INTERSPEECH*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Elisa Perego, F. Del Missier, M. Porta, and M. Mosconi. 2010. The cognitive effectiveness of subtitle processing. *Media Psychology*, 13:243––272.

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, October. Association for Computational Linguistics.

Dhevi J. Rajendran, Andrew T. Duchowski, Pilar Orero, Juan Martínez, and Pablo Romero-Fresco. 2013. Effects of text chunking on subtitling: A quantitative and qualitative examination. *Perspectives*, 21(1):5–21.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796, Online, July. Association for Computational Linguistics.

Pablo Romero-Fresco, 2010. *Standing on quicksand: hearing viewers' comprehension and reading patterns of respoken subtitles for the news*, pages 175 – 194. Brill, Leiden, The Netherlands.

Pablo Romero-Fresco. 2011. *Subtitling through speech recognition: Respeaking*. Manchester: St. Jerome.

Frederick W. M. Stentiford and Martin G. Steer. 1988. Machine Translation of Speech. *British Telecom Technology Journal*, 6(2):116–122.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Alex Waibel, Ajay N. Jain, Arthur E. McNair, Hiroaki Saito, Alexander G. Hauptmann, and Joe Tebelskis. 1991. JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP 1991*, pages 793–796, Toronto, Canada, May 14-17.

Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Liang Lu, Guoli Ye, and Ming Zhou. 2020. Low latency end-to-end streaming speech recognition with a scout network. *arXiv preprint arXiv:2003.10369*.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Proceedings of Interspeech 2017*, pages 2625–2629, Stockholm, Sweden, August.

Hao Xiong, Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Dutongchuan: Context-aware translation model for simultaneous interpreting. *arXiv preprint arXiv:1907.12984*.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474, Online, August. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China, November. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy, July. Association for Computational Linguistics.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019c. Speculative beam search for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1395–1402, Hong Kong, China, November. Association for Computational Linguistics.

Baigong Zheng, Kaibo Liu, Renjie Zheng, M. Ma, Hairong Liu, and L. Huang. 2020. Simultaneous translation policies: From fixed to adaptive. *ArXiv*, abs/2004.13169.