UNIVERSITÀ
DI TRENTO
Department of
Information Engineering and Computer Science

Doctoral School in
Information and Communication Technology

Direct Speech Translation in Constrained Contexts: the Simultaneous and Subtitling Scenarios

Sara Papi

Advisors

Marco Turchi	Zoom Video Communications
Matteo Negri	Fondazione Bruno Kessler

Committee

Claudio Fantinuoli	University of Mainz & KUDO Inc.
Juan Pino	Meta AI

January 2024

Abstract

This PhD thesis summarizes the results of a three-year comprehensive investigation into the dynamic domain of speech translation (ST), with a specific emphasis on application scenarios requiring adherence to the additional constraints posed by simultaneous speech translation and automatic subtitling. These additional constraints, revolving around aspects such as latency and on-screen spatio-temporal conformity, add layers of complexity and thereby complicate the inherent challenges of ST. I started my exploration with a novel paradigm, direct speech translation, which was in its early stages during the beginning of my journey. Along this direction, in the pursuit of advancing simultaneous ST (SimulST), my research challenged the conventional approach of creating task-specific direct architectures. Instead, the focus was on leveraging the intrinsic knowledge acquired by offline-trained direct ST models for simultaneous inference. A pivotal contribution of this endeavor was the finding that offline-trained ST systems can not only compete with but potentially surpass the quality and latency of those specifically trained for simultaneous scenarios. An important subsequent step has been taken by leveraging cross-attention information extracted from an offline direct ST model for SimulST, demonstrating its potential to deliver high-quality, low-latency translations with minimal computational costs and thus achieve an optimal balance between translation quality and latency. The exploration of automatic subtitling delved into the complexities of spatio-temporal constraints, highlighting the interplay between translation quality, text length, and display duration. The recognition of the importance of prosody and speech cues shaped the development of direct architectures for the task. Relevant findings include the effectiveness of a multimodal segmenter, leveraging both audio and textual cues for optimal segmentation into subtitles. Furthermore, my research showcased the capability of direct ST models to generate complete subtitles, offering translations appropriately segmented with corresponding timestamps, and demonstrating competitive performance against existing cascaded production tools. In conclusion, the insights gleaned in this PhD from both fields mark substantial technological progress, which I believe will set the stage for the wide adoption of direct ST systems in the two challenging domains of simultaneous ST and automatic subtitling.

Keywords

speech, translation, direct speech translation, constraints, simultaneous, subtitling

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

The globalization of business, education, and entertainment has moved human interaction to the online sphere, opening a new era characterized by virtual meetings,[1] e-learning platforms,[2] and worldwide digital content consumption.[3] The boom in online communication[4] is setting new challenges for achieving barrierless interaction among users with diverse linguistic and accessibility requirements (Kožuh and Debevc, 2018; Abarca et al., 2020; Dyzel et al., 2020).

As individuals who speak different languages engage in digital interaction, the need for effective language translation becomes increasingly critical.[5] Speech-to-text translation (ST) emerges as a pivotal technology in this context (Stentiford and Steer, 1988; Waibel et al., 1991), bridging linguistic gaps and facilitating communication, consequently enhancing accessibility and inclusivity, fostering collaboration, knowledge exchange, and cultural understanding (Takezawa et al., 1998; Black et al., 2002; Waibel, 2004; Besacier et al., 2006; Fügen, 2009; Bansal et al., 2017; Anastasopoulos and Chiang, 2017;

[1]Workers are spending an average of 20 hours a week using digital communication tools (`https://www.forbes.com/advisor/business/digital-communication-workplace/`).

[2]77% of corporations in the US use online learning tools, such as Udemy and Coursera, and the e-learning market is forecasted to grow to $320 billion by 2025 (`https://codeless.co/elearning-statistics/`).

[3]In 2023, the number of internet users attested to 5.18 billion, meaning that two-thirds of the global population is currently connected to the world wide web (`https://www.statista.com/topics/1145/internet-usage-worldwide/`).

[4]The number of social media users worldwide is constantly increasing and is expected to reach 5.17 billion in 2024 (`https://www.statista.com/topics/1164/social-networks/`).

[5]The market size of the Translation Services industry is $10.3 billion in 2023, an increase of 6.09% from 2022 (`https://www.ibisworld.com/industry-statistics/market-size/translation-services-united-states/`).

Dessloch et al., 2018; Lommel, 2018; Bano et al., 2020; Lee et al., 2022; Salesky et al., 2023, among others).

Unlike traditional *text-based* machine translation methods (Zens et al., 2002; Koehn et al., 2003), ST aims to seamlessly convert *spoken* words from one language into another, often in real-time, providing a more natural way to understand language. However, the development and deployment of effective ST systems are not without challenges: the inherent complexities of speech, encompassing variations in accents, speaking rates, disfluencies, and background noise, entail significant obstacles (Derwing and Munro, 2009; Salesky et al., 2019; Sperber and Paulik, 2020). Furthermore, in specific scenarios where constraints like time (e.g., output latency), space (e.g., characters to be displayed on the screen), computational resources (e.g., the necessity of running models only on CPUs), or limited data availability (e.g., low resource language) must be considered, delivering reliable, and high-quality translation systems becomes even more complicated (Bansal et al., 2018; Matusov et al., 2019; Ren et al., 2020).

This thesis delves into the domain of ST, focusing on the latest research direction where neural architectures (LeCun et al., 2015; Sejnowski, 2018) are trained to directly generate the desired translation from the input speech without any intermediate steps (Bérard et al., 2016; Weiss et al., 2017). By exploring novel approaches, techniques, and frameworks, the primary objective of this PhD was to push the boundaries of current ST capabilities, particularly in the context of specific use-case scenarios where additional constraints come into play. In the following section, we delve deeper into these constraints, isolating the specific requirements for the specific applications analyzed in this PhD thesis: simultaneous ST and automatic subtitling. This exploration gave me the possibility to establish clear and specific goals for my research path.

1.2 Research Questions

The core objective of conventional ST systems is to achieve the utmost **quality** of automatic textual translations (Callison-Burch et al., 2006). However, the landscape of ST applications encompasses tasks in which achieving high translation quality alone is not sufficient. As addressed throughout this PhD thesis, when confronted with additional constraints, the fundamental challenge for the ST systems becomes finding the right balance between optimizing translation quality and ensuring the fulfillment of specific constraints.

In this context, two constrained scenarios were analyzed during this PhD due to

their inherent scientific and industrial interest: **simultaneous speech translation** (Section 1.2.1) and **automatic subtitling** (Section 1.2.2). In the remainder of this section, we shortly introduce the two tasks, highlighting their peculiarities and inherent challenges.

1.2.1 Simultaneous ST

In real-time[6] or simultaneous ST, the primary constraint (Figure 1.1) is dictated by the need to minimize the **latency** required to deliver the output (Sarkar, 2016), which represents the time delay from when an utterance is spoken in the source language to when it is translated into the target language. This temporal constraint introduces additional challenges in the production of the final translation (Huang et al., 2020). On one hand, the translated text has to be promptly displayed, ideally contemporaneously with the spoken words, aligning smoothly with the natural pace of the speech and its alternation with pauses and non-speech events. For example, limits on the latency acceptability have been set between 2 and 6 seconds for the ear-voice-span[7] under different conditions and language pairs (Yagi, 2000; Chmiel et al., 2017; Fantinuoli and Prandi, 2021). On the other hand, the translation has to uphold a commendable standard of quality, maintaining a level of linguistic accuracy and coherence to enable a comprehensive understanding of the conveyed message for the end user (Macháček et al., 2023a). Therefore, finding a trade-off between translation quality and latency is essential for a positive user experience (Niehues et al., 2018b). This balance ensures that the translation is adequately synchronized, allowing users to efficiently process and comprehend the content, and seamlessly understand or participate in the ongoing conversation.

Despite the rising demand for real-time technologies,[8] current systems struggle to achieve good performance (Zhang et al., 2022b), that is an optima quality-latency trade-off. Moreover, current SimulST models are also characterized by long and complicated training procedures (e.g., computing complex training losses or performing multiple training stages) to optimize the ST models for the simultaneous task (Liu et al., 2021b;

[6]Real-time is a broad term (Seligman, 1997) that refers to "the very short amount of time needed for computer systems to receive data and information and then communicate it or make it available" (`https://dictionary.cambridge.org/dictionary/english/real-time`).

[7]The ear-voice-span (EVS), also known as "décalage", refers to the delay between the original speaker and the interpreter in simultaneous interpretation or translation.

[8]The Global Real-Time Language Translation Device market is anticipated to rise at a considerable rate (CAGR) of USD million between 2023 and 2030 (`https://www.precisionreports.co/enquiry/request-sample/21640680`).

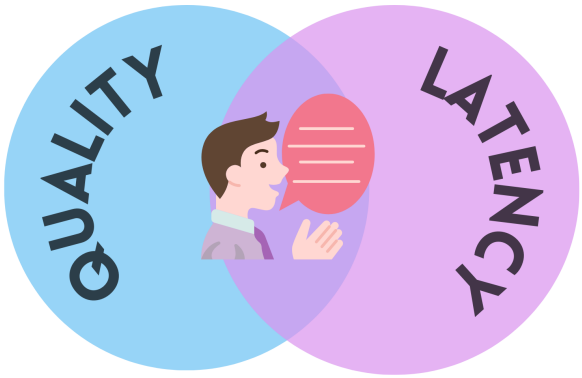


Figure 1.1: Constraints of simultaneous ST: translation quality and latency (i.e., the time delay from when an utterance is spoken in the source language to when it is translated into the target language).

Zaidi et al., 2021; Chang and Lee, 2022; Zhang and Feng, 2022; Omachi et al., 2023), which sometimes also involve creating and maintaining several models to accommodate different latency requirements (Ren et al., 2020; Ma et al., 2020b; Zeng et al., 2021).

In light of these major drawbacks, my objective during this PhD journey was to reassess the development of simultaneous ST systems from a different perspective. Rather than devising sophisticated and ad-hoc training procedures for the task, I focused on analyzing the already existing ST systems to understand if they possess intrinsic knowledge that could be leveraged for real-time applications. Specifically, the research questions guiding this investigation were twofold:

1. **Are these sophisticated and ad-hoc training procedures necessary for the simultaneous ST task?**

2. **Can we exploit the knowledge already acquired by offline direct ST models to guide them during the simultaneous inference?**

Therefore, the main goal of my studies on simultaneous ST (Chapter 3) was to investigate the feasibility and potential extent to which a standard direct offline ST model could be repurposed for real-time scenarios.

1.2.2 Automatic Subtitling

Subtitling is the process of providing short pieces of text translating the content of spoken dialogue in audiovisual media, such as movies, video lectures, and TV shows.

Therefore, providing automatic subtitles (automatic subtitling) is a multifaceted task characterized by several spatio-temporal constraints (Figure 1.2) related to when and how the subtitles have to be displayed on the screen (Cintas and Remael, 2021).

Spatial or **length** constraints, primarily associated with subtitle length, play a crucial role (Michael P. Hinkin and Miranda, 2014). Excessively long subtitles can present cognitive challenges for viewers attempting to process them while concurrently watching the video (Szarkowska et al., 2011). Conversely, overly short and compressed subtitles (Burnham et al., 2008; Szegedy et al., 2016) might entail a loss of information, which introduces difficulties in accurately interpreting the spoken content (Chen and Ho, 2022).

Temporal or **duration** constraints revolve around the synchronization of subtitles with the audiovisual content, an indispensable aspect of user experience and comprehension (Bisson et al., 2014; Szarkowska and Gerber-Morón, 2018). Subtitles must strike a delicate balance, staying on the screen long enough for users to easily read them, yet not lingering excessively to avoid disrupting the natural flow of the content (Kruger et al., 2022). Achieving this synchronicity is pivotal not only for enhancing user understanding but also sustaining a pleasant engagement with the video (Perego, 2008; Szarkowska and Gerber-Morón, 2018).

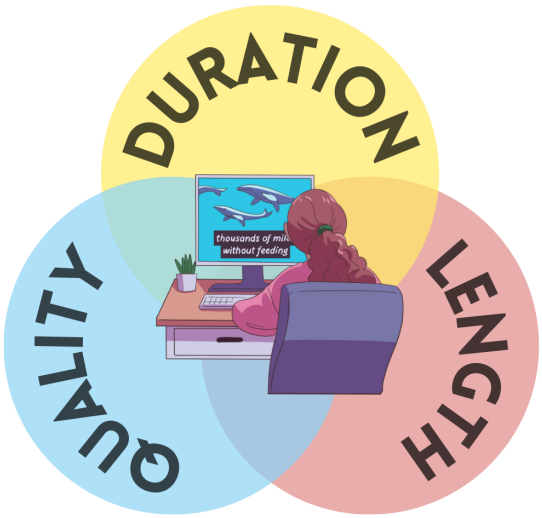


Figure 1.2: Constraints of automatic subtitling: translation quality, length conformity (i.e., not excessively short or long subtitles to ease user comprehension), and duration conformity (i.e., synchronized subtitles that stay on-screen enough to let the user understand their content).

Being a fundamental part of human spoken communication, prosody (Hirschberg, 2006) and speech cues in general (e.g., pauses, and hesitations) are important elements,

which can help with subtitle segmentation and temporization (Karakanta et al., 2020a, 2021a). Several studies proved the importance of prosody and attempted to integrate this information into the process of creating subtitles (Öktem et al., 2019; Federico et al., 2020; Virkar et al., 2021; Tam et al., 2022; Effendi et al., 2022), which was realized only, at that time, using a pipeline of several models, losing direct contact with this important property.

In light of this, direct models emerge as a potential solution, capable of seamlessly leveraging such informational cues. Building upon this intuition, my research during this PhD focused on the strategic utilization of direct models for subtitle segmentation and fully automatizing the subtitling task. Specifically, the research questions are:

1. **Is there a way to exploit prosody and speech cues accessible by direct systems to build automatic subtitling datasets starting from already existing ST corpora?**

2. **Is it possible to exploit a direct ST model for producing full subtitles (translated texts with their corresponding timestamp)?**

Initially, I explored the use of direct multimodal approaches, capable of exploiting text and audio, to create a tool for automatically segmenting translated texts into appropriately timed subtitles. Subsequently, I worked on developing the first direct ST model employed for generating complete subtitles – translations properly segmented with corresponding timestamp information – with the ultimate goal of positioning this model as a competitive alternative to existing state-of-the-art cascade approaches (Chapter 4).

1.3 List of Contributions

In this section, all the contributions that I made during my PhD are listed in chronological order (most recent first).

The contributions are grouped according to the specific research questions addressed and, at the highest level, are divided into three parts: Simultaneous ST and Automatic Subtitling, which are the main pillars of this thesis, and other contributions, which are the works done, both as first author or in collaboration with other researchers, in the field of ST. The "*" symbol indicates equal contributions.

1.3.1 Simultaneous ST

- How can a unified model be developed to concurrently generate transcriptions and translations in a streaming scenario, and what diverse approaches can be explored to optimize the delivery of both outputs?
  - Papi, S., Wang, P., Chen, J., Xue, J., Kanda, N., Li, J., Gaur, Y. (2023). "Leveraging Timestamp Information for Serialized Joint Streaming Recognition and Translation". In Proceedings of ICASSP 2024.[9]
  - Papi, S., Wang, P., Chen, J., Xue, J., Li, J., Gaur, Y. (2023). "Token-Level Serialized Output Training for Joint Streaming ASR and ST Leveraging Textual Alignments". In Proceedings of ASRU 2023.[10]
- How can attention mechanisms be effectively utilized to guide a speech translation model in simultaneous inference?
  - Papi, S., Turchi, M., Negri, M. (2023). "AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation". In Proceedings of INTERSPEECH 2023.
  - Papi, S., Negri, M., Turchi, M. (2023). "Attention as a Guide for Simultaneous Speech Translation". In Proceedings of ACL 2023. **Best paper nomination.**
- How can we adapt the existing simultaneous speech translation metrics to deal with scenarios where the predicted output exceeds the length of the corresponding gold reference?
  - Papi, S., Gaido, M., Negri, M., Turchi, M. (2022). "Over-Generation Cannot Be Rewarded: Length-Adaptive Average Lagging for Simultaneous Speech Translation". In Proceedings of the Third Workshop on Automatic Simultaneous Translation.
- Is it feasible to leverage a direct speech translation system for simultaneous tasks without re-training or adaptation? How does this approach perform across various architectures and data conditions?
  - Papi, S., Gaido, M., Negri, M., Turchi, M. (2022). "Does Simultaneous Speech Translation need Simultaneous Models?". In Findings of EMNLP 2022.

[9]Work done during an internship at Microsoft.
[10]Work done during an internship at Microsoft.

  - Gaido*, M., Papi*, S., Fucci, D., Fiameni, G., Negri, M., Turchi, M. (2022). "Efficient yet Competitive Speech Translation: FBK@IWSLT2022". In Proceedings of IWSLT 2022.

- To what extent do visualization modalities impact the delivery of simultaneous speech translation outputs, and how significant is their role in enhancing comprehension and user experience?

  - Papi, S., Negri, M., Turchi, M. (2021). "Visualization: The missing factor in simultaneous speech translation". In Proceedings of the Eighth Italian Conference on Computational Linguistics.

1.3.2 Automatic Subtitling

- Can we fully automatize the subtitling process by leveraging a direct ST model? How does it perform compared to state-of-the-art approaches and production tools in different languages and data conditions?

  - Papi, S., Gaido, M., Karakanta, A., Cettolo, M., Negri, M., Turchi, M. (2023). "Direct Speech Translation for Automatic Subtitling". In Transactions of the Association for Computational Linguistics.
  - Papi, S., Gaido M., Negri, M. (2023). "Direct Models for Simultaneous Translation and Automatic Subtitling: FBK@IWSLT2023". In Proceedings of IWSLT 2023.

- Can a multimodal (speech and text) direct model effectively segment text into subtitles? To what extent can this model be integrated "as is" into a subtitling pipeline, and how does it perform for data augmentation?

  - Papi, S., Karakanta, A., Negri, M., Turchi, M. (2022). "Dodging the Data Bottleneck: Automatic Subtitling with Automatically Segmented ST Corpora". In Proceedings of AACL-IJCNLP 2022.

- Is it feasible to produce subtitles in real-time, and what are the implications of various visualization strategies on the quality and user experience of the generated subtitles?

  - Karakanta*, A., Papi*, S., Negri, M., and Turchi, M. (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).

1.3.3 Other contributions

- What is the significance of software quality within and beyond the NLP community, and how can the repercussions of neglecting software quality in research be empirically highlighted? What countermeasures can be introduced to enhance code correctness?

  – Papi*, S., Gaido*, M., Pilzer, A., Negri, M. (2023). "When Good and Reproducible Results are a Giant with Feet of Clay: The Importance of Software Quality in NLP". Under review.

- How can we integrate information about masculine/feminine forms to use into ST systems without retraining the model?

  – Fucci, D., Gaido, M., Papi, S., Cettolo, M., Negri, M., Bentivogli, L. (2023). "Integrating Language Models into Direct Speech Translation: An Inference-Time Solution to Control Gender Inflection". In Proceedings of EMNLP 2023.

- Is it possible to jointly produce speech translations and tagged named entities within a single model and in real-time?

  – Gaido, M., Papi, S., Negri, M., Turchi, M. (2023). "Joint Speech Translation and Named Entity Recognition". In Proceedings of INTERSPEECH 2023.

- Is it possible to build an ST model without without the initial subsampling of the audio? What advantages does this design offer in comparison to typical speech processing architectures?

  – Papi*, S., Gaido*, M., Negri, M., Turchi, M. (2021). "Speechformer: Reducing Information Loss in Direct Speech Translation". In Proceedings of EMNLP 2021.

- In the development of competitive offline ST systems, how can the mismatch between training and testing conditions be effectively addressed?

– Papi, S., Gaido, M., Negri, M., Turchi, M. (2021). "Dealing with training and test segmentation mismatch: FBK@ IWSLT2021". In Proceedings of IWSLT 2021.

Chapter 2

Preliminaries

2.1 Direct Speech Translation

Speech-to-text translation or, more simply, speech translation (ST) involves the conversion of spoken sentences from one language into written text in another language. The relevance of this area is underscored by its increasing integration into various aspects of our daily activities and its versatility in addressing a range of applications, including translating lectures (Fügen, 2009; Dessloch et al., 2018) and conferences (Salesky et al., 2023), facilitating travel conversations (Takezawa et al., 1998), easing communication with unwritten languages and dialects (Besacier et al., 2006; Lee et al., 2022), documenting endangered languages (Bansal et al., 2017; Anastasopoulos and Chiang, 2017), as well as supporting humanitarian expeditions and health (Black et al., 2002; Munro, 2010; Martin et al., 2015).

In its initial stages, ST was tackled through modular solutions, called *cascade* architectures, employing dedicated systems for distinct subtasks (Stentiford and Steer, 1988; Waibel et al., 1991). Typically, these architectures comprised two main modules: an automatic speech recognition (ASR) system responsible for generating the transcripts of spoken utterances, and a machine translation (MT) system designed to translate the predicted transcripts into the desired target language.

Due to their modular nature, cascade architectures have high adaptability across languages and domains but, at the same time, face well-known challenges associated with concatenating multiple systems. These challenges include: *i)* the need for ad-hoc training and maintenance procedures for the ASR and MT modules (Peitz et al., 2012; Ruiz et al., 2017; Martucci et al., 2021), *ii)* the propagation of errors from the transcription to the translation step (Sperber and Paulik, 2020), *iii)* the loss of speech information

(e.g., prosody) in the transcriptions that might be useful to produce the translations (Tam et al., 2022), and *iv)* an increased latency due to sequential execution by two modules (Weller et al., 2021).

For the aforementioned reasons, end-to-end or *direct* models have become increasingly popular thanks to their potential to execute the whole task without relying on intermediate representations (Bérard et al., 2016; Weiss et al., 2017). However, the direct execution of translation from speech poses more complex challenges and, in its initial proposal in 2016, despite its promising potential, this approach faced a considerable performance gap compared to cascade models (Niehues et al., 2018a, 2019).

Notably, in recent times, this performance gap has been steadily diminishing (Ansari et al., 2020; Bentivogli et al., 2021; Anastasopoulos et al., 2021, 2022; Agarwal et al., 2023) and the ST landscape has evolved, witnessing a surge in the development of direct systems designed to tackle multiple subtasks, including offline (Xie, 2023; Zhou et al., 2023; Huzaifah et al., 2023), simultaneous (Yan et al., 2023; Polák et al., 2023), multilingual (Gow-Smith et al., 2023; Wang et al., 2023b), low resource (Kesiraju et al., 2023; Mbuya and Anastasopoulos, 2023; Kesiraju et al., 2023), subtitling (Papi et al., 2023b; Bahar et al., 2023), and dialect translation (Deng et al., 2023; Radhakrishnan et al., 2023; Laurent et al., 2023). This evolution underscores the increasing viability and competitiveness of end-to-end models in addressing the challenges of ST, which has made them an interesting research direction in the last few years (Xu et al., 2023).

In light of the recent advances in direct ST, during my PhD, I focused on this promising paradigm and on its application to specific sub-tasks, simultaneous ST, and automatic subtitling, which combine the general challenges of translating speech with the specific ones rising from these two application domains. In the following, before delving into my contributions to the specific areas of Simultaneous ST (Chapter 3) and Automatic Subtitling (Chapter 4), I provide the fundamental notions to better understand the inherent challenges, the technology solutions, and the evaluations discussed in this thesis. First, I present the two main architectures used in direct speech processing, Transformer and Conformer (Section 2.2), then I discuss the data used for direct ST model training, including corpora and data augmentation techniques (Section 2.3). Lastly, I present the metrics used for the evaluation of the performance of ST systems (Section 2.3).

2.2 Architectures

In the following, I present two key sequence-to-sequence architectures widely utilized in speech processing: Transformer (Section 2.2.1) and Conformer (Section 2.2.2). Both these architectures were used during my PhD, with the Conformer replacing the Transformer in my latest works.

2.2.1 Transformer

Transformer (Vaswani et al., 2017) is the most widely used architecture in speech processing (Latif et al., 2023). It is composed of an encoder, which is in charge of processing the input sequence to find an internal representation, and a decoder, which produces the output exploiting the encoded information.

This encoder-decoder architecture relies on the *attention* mechanism (Bahdanau et al., 2015), which allows any vector of the input sequence to access any part of the output sequence regardless of its position. The adopted attention mechanism is a variant of dot-product attention (Luong et al., 2015), which, starting from the query vector Q, the key vector K, and the value vector V, is formulated as:

$$\text{Attn}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.1}$$

where d_k is the size of the key vector.

In Transformer models, there are two types of attention: *self-attention* and *cross-attention* (or encoder-decoder attention). In self-attention, Q, K, and V are derived from the same input sequence X, transformed by linear projections (W_Q, W_K, and W_V) with learned weights:

$$\text{SelfAttn}(X) = \text{softmax}\Big(\frac{W_Q X (W_K X)^T}{\sqrt{d_k}}\Big)W_V X \tag{2.2}$$

In cross-attention, K, and V are derived from the encoder output $E(X)$ while Q from the previous decoder output Y:

$$\text{CrossAttn}(X, Y) = \text{softmax}\Big(\frac{W_Q E(X) (W_K Y)^T}{\sqrt{d_k}}\Big)W_V E(X) \tag{2.3}$$

To enable parallel computation, Vaswani et al. (2017) introduced *multi-head* attention, dividing Q, K, and V into h heads, transforming each with dedicated weight matrices,

computing attention separately, and concatenating the results as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Attn}(q_0, k_0, v_0), ..., \text{Attn}(q_h, k_h, v_h))W_O \quad (2.4)$$

where $W_O \in \mathbb{R}^{(d_k, d_k)}$ is a learned matrix. As multi-head attention is always employed in current architectures, henceforth, the term "attention" refers to the multi-head variant.

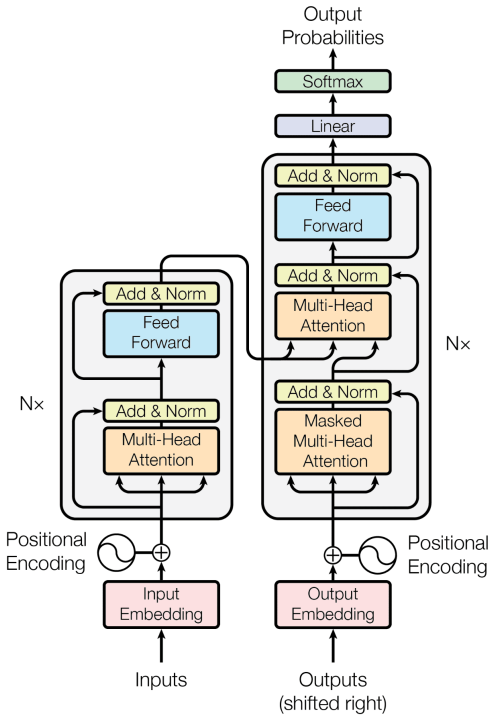


Figure 2.1: Transformer architecture. Credits to (Vaswani et al., 2017).

The final Transformer architecture (Figure 2.1) consists of a sequence of Transformer encoder layers, each incorporating self-attention, and a series of Transformer decoder layers, encompassing both self-attention and cross-attention mechanisms. Within each encoder layer, there is a combination of attention and a linear layer or feed-forward network (FFN), with both components followed by layer normalization (Ba et al., 2016) and augmented by residual connections (He et al., 2016). Analogously, each decoder layer encompasses self-attention and an FFN layer, with the addition of cross-attention placed in between these components.

While the Transformer architecture has immediately demonstrated success in MT, language modeling, and natural language processing in general (Radford et al., 2018; Devlin et al., 2019), its direct application to speech processing encountered challenges due to prohibitive memory requirements. Specifically, self-attention layers have quadratic

memory complexity with respect to the length of the input sequence (T), as the QK^T product with $Q, K \in \mathbb{R}^{(T,d)}$ generates a matrix of dimension $T \times T$.

Given that speech input sequences are generally around 10 times longer than their corresponding textual counterpart, this quadratic memory complexity poses a hurdle to the straightforward application of the Transformer without facing memory issues. To address this, Dong et al. (2018) and Bérard et al. (2018) proposed leveraging convolutional neural networks (CNNs – LeCun 1989), either in 1-dimensional (Wang et al., 2020a) or 2-dimensional (Di Gangi et al., 2019c) configurations, to reduce the input sequence length. In practice, the input sequence undergoes initial processing with two layers of CNN, each having a subsampling factor of 2 and resulting in an overall subsampling factor of 4. Subsequently, with its reduced dimensions, the sequence is processed using the Transformer architecture. Due to its success, this approach has become the *de facto* strategy adopted in subsequent speech-processing architectures, including the state-of-the-art Conformer model (Section 2.2.2).

2.2.2 Conformer

Unlike Transformer, which is a general-purpose architecture, the Conformer architecture (Gulati et al., 2020) was specifically proposed for speech processing tasks. The modifications proposed with this architecture, as compared to the Transformer, are focused on the structure of the encoder layers, as shown in Figure 2.2.

In particular, the Conformer encoder layer introduces several key improvements:

- **Relative Sinusoidal Positional Encodings**: to facilitate improved generalization across varied input lengths, relative sinusoidal positional encodings are incorporated into the self-attention mechanism (Dai et al., 2019);
- **Two Half-Dimensional Feed-Forward Networks**: instead of a single FFN layer, the Conformer utilizes two half-dimensional FFNs that encapsulate the self-attention mechanism. This design choice is inspired by the Macaron-Net architecture (Lu et al., 2019);
- **Additional Convolutional Module**: a Convolutional Module is placed immediately after the self-attention layer and before the final FFN layer, introducing a novel element to the Conformer architecture.

The Convolutional Module (Figure 2.3) applies a sequential series of operations to the input. Initially, the input undergoes layer normalization (LayerNorm) and a

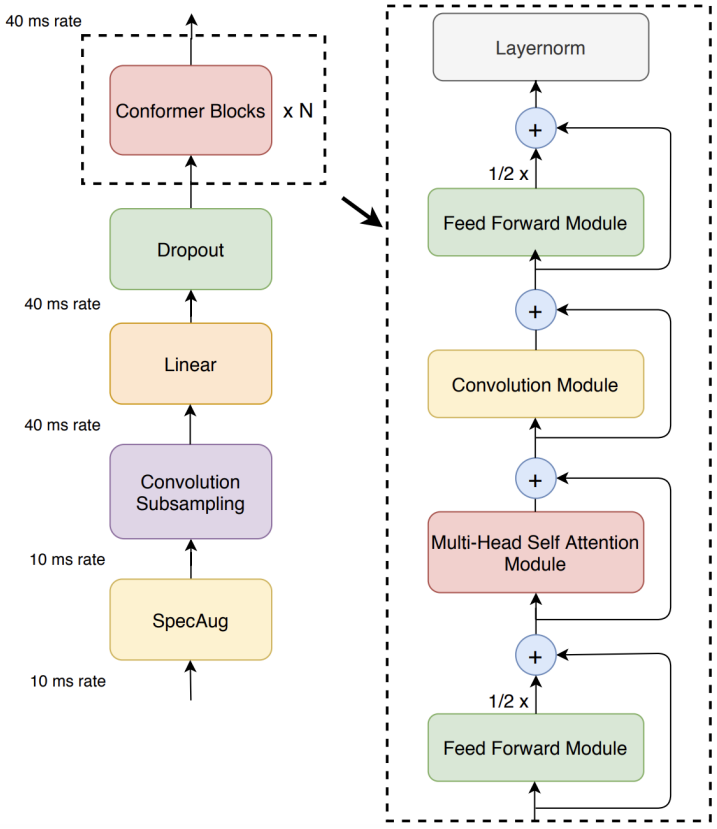


Figure 2.2: Conformer Encoder. Credits to (Gulati et al., 2020).

point-wise convolution, which doubles the size of the input features. Subsequently, a Gated Linear Unit (GLU) activation function (Dauphin et al., 2017) is employed to restore the features to their original size. Next, a depth-wise convolution with a kernel size of 31 is applied, followed by batch normalization (BatchNorm – Ioffe and Szegedy 2015), the Swish activation function (Ramachandran et al., 2017), another point-wise convolution, and a Dropout module (Srivastava et al., 2014) that randomly masks a percentage of features to mitigate overfitting. Lastly, the entire Convolutional Module is encapsulated in a residual connection.

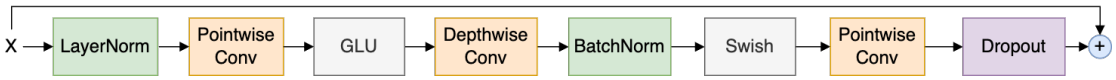


Figure 2.3: Convolutional Module of the Conformer Encoder.

The Conformer architecture, initially designed for ASR, has demonstrated cutting-edge performance in the ST domain (Guo et al., 2021). Its excellence is underscored by its widespread adoption, as evidenced by its notable milestone in the number of citations, surpassing 2,000 in the year 2023. This acclaim has translated into its integration into

various recent works across the spectrum of speech processing research, as observed in recent publications (Ma et al., 2021a; Srivastava et al., 2022; Li and Doddipatla, 2023), also in ST (Inaguma et al., 2021a).

During my PhD, I adopted both architectures: during the initial phase of my research, discussed in Sections 3.2.1 and 4.2.1, I adopted the Transformer while, for all the subsequent works (Sections 3.2.2, 3.2.3, and 4.2.2), I adopted the emerging and better performing Conformer. As regards their inner attention mechanism, its use is at the core of my latest contributions on simultaneous ST (Sections 3.2.2, and 3.2.3).

2.3 Data

Being direct ST a relatively recent field of investigation, one of the primary challenges encountered in its early stages was the scarcity of data. At the beginning of this PhD project, there were only three existing ST datasets covering multiple languages, namely:

- **MuST-C (Di Gangi et al., 2019a):** it contains about 500 hours of TED talks[1] in English with translations into 8 languages, later extended to 14 target languages (Cattoni et al., 2021);
- **EuroParl-ST (Iranzo-Sánchez et al., 2020)**: it contains debates carried out in the European Parliament in the period between 2008 and 2012 and comprises 30 different translation directions from and into 6 European languages;
- **CoVoST-2 (Wang et al., 2020b)**: it is based on the CommonVoice ASR corpus (Ardila et al., 2020) and covers translations from English into 15 languages and from 21 languages into English, with a total of 2,880 hours;

These datasets consist of audio recordings in the source language, their corresponding textual translations into multiple languages, and, optionally, transcriptions in the source language. Despite having three dataset options, I predominantly utilized the MuST-C dataset throughout this PhD (Chapters 3, and 4) as it represented the standard benchmark in the field of ST, facilitating comparisons with existing works.

In response to the data scarcity problem represented by the limited availability of multilingual corpora, the research community has also actively pursued innovative approaches in both **data augmentation** (Section 2.3.1) and **knowledge transfer** (Section 2.3.2) techniques, which I expand upon in the following.

[1] `https://www.ted.com/talks`

2.3.1 Data Augmentation

Data augmentation techniques play a pivotal role in enhancing the performance and robustness of models, including direct ST. In this context, where the availability of diverse and extensive datasets is often limited, data augmentation becomes a key strategy to mitigate the data scarcity issue. By introducing variations and diversity into the training data, these techniques aim to expose the model to a more comprehensive range of possible inputs, making it more suitable for handling diverse linguistic nuances, speech patterns, and environmental conditions.

SpecAugment. The most popular data augmentation technique is *SpecAugment*, initially introduced for ASR by Park et al. (2019), which has also demonstrated its effectiveness in ST (Bahar et al., 2019). The concept behind SpecAugment is to modify the audio features representing speech, enhancing the variability of training data and contributing to the development of more robust systems. Operating on input features, SpecAugment is applied with a probability p and involves masking (zeroing out) consecutive portions of the input in both the frequency and time dimensions.

Time Stretch. With a similar objective to SpecAugment, Nguyen et al. (2020b) proposed the *time stretch*, which directly manipulates audio features with the aim of achieving effects similar to *speed perturbation* (Ko et al., 2015) to enhance system robustness against variations in speech rate. This approach involves segmenting the input sequence into windows of w features and subsequently resampling each window using a random factor s drawn from a uniform distribution.

Synthetic Data. Another commonly utilized approach to cope with data scarcity involves the generation of *synthetic data*, either in textual or audio format. To leverage the availability of large ASR datasets (Ardila et al., 2020; Wang et al., 2021), parallel audio-translation pairs are created by translating the transcript of each audio using an MT model (Jia et al., 2019). This method can also be viewed as a knowledge transfer technique known as sequence-level knowledge distillation (sequence-level KD) since it transfers (or distills) the knowledge of the MT model into the ST model (Gaido et al., 2021b, 2022d). Synthetic audio creation, instead, was initially explored in ASR (Mimura et al., 2018; Li et al., 2018; Rossenbach et al., 2020). This technique involves generating speech using text-to-speech models starting from gold or back-translated transcriptions. While this approach is less common in the context of direct ST (Lam et al., 2022), as it

demands more time and resources for the creation of artificial audio, the generation of synthetic translations remains a well-established practice (Di Gangi et al., 2019b; Gaido et al., 2020b; Inaguma et al., 2021b).

Throughout this PhD, I adopted the SpecAugment data augmentation technique, which was also applied by default by the deep learning library, *fairseq s2t* (Wang et al., 2020a), on which I based all my experiments. Moreover, for some works (Sections 3.2.1, 3.2.2, 3.2.3, and 4.2.2), I also employed sequence-level KD to enhance the translation quality, as it is one of the less resource expensive technique to apply since it stores only additional (translated) texts.

2.3.2 Knowledge Transfer

The notion of knowledge transfer in neural networks mirrors human learning, involving the transmission of information acquired by a neural network trained on a specific task to another neural network, usually smaller, which may be designed for the same or a different task (Gutstein et al., 2008). This term is also used when systems trained on multiple tasks or output modalities (e.g., multilingual models) can transfer the acquired knowledge from one task/modality to the others (Escolano et al., 2019; Dabre et al., 2020). The set of these techniques, the main examples of which will be described below, can be effectively employed alone or in conjunction with the aforementioned data augmentation techniques.

Model pre-training. In the context of direct ST, knowledge transfer from high-resource tasks is traditionally implemented through *model pre-training*, which consists of initializing the ST model, or part of it, with the weights of ASR or MT models having the same structure but trained on a larger amount of data. Studies by Bérard et al. (2018) and Bansal et al. (2019) have demonstrated the effectiveness of initializing the ST encoder with the weights of an ASR model trained on extensive ASR corpora. However, initializing the ST decoder with an MT model alone has shown limited effectiveness (Bansal et al., 2019), unless supplemented with an adapter layer (Bahar et al., 2019). Despite recent work suggesting some advantages (Li et al., 2021), the performance benefits of this approach still remain unclear.

Multitask learning. Another strategy for knowledge transfer in direct ST is *multitask learning*. In this approach, a single shared encoder is utilized by two separate decoders, each dedicated to generating transcripts and translations (Weiss et al., 2017).

Building on this concept, Anastasopoulos and Chiang (2018) introduced a variation that enables each decoder to attend to the representations generated by its counterpart. Alternatively to adding a separate ASR decoder, Bahar et al. (2019) proposed to leverage the Connectionist Temporal Classification (CTC – Graves et al. 2006) as an auxiliary loss to predict transcriptions directly from the encoder output (Kim et al., 2017). The CTC loss enables the generation of output sequences of variable length, which is crucial in ST, where input sequences (audio) are typically longer than output sequences (transcriptions). At each time step, the CTC produces a probability distribution over possible target tokens, incorporating a dedicated `<blank>` symbol indicating the absence of a target value. These distributions are then employed to calculate probabilities for different sequences, collapsing consecutive equal predictions and removing `<blank>` symbols. Lastly, the resulting sequences are compared with the target sequence (transcription).

Knowledge Distillation. The last commonly adopted knowledge transfer technique is knowledge distillation (KD), which was introduced to transfer knowledge from a big model into a small, compressed one (Hinton et al., 2015). The objective is to have a small model (referred to as the *student* in the KD learning procedure) that is trained to mimic the probability distribution of its larger counterpart (referred to as the *teacher*) when processing the same input, so as to achieve comparable performance to the teacher. This involves using the probabilities generated by the teacher as a reference during the training of the student, rather than the usual reference distribution where probability 1 is assigned to the correct label and all others are set to 0. In practical terms, this means that the student is not only optimized for the cross-entropy loss function but also to minimize the distance between its probability distribution and that generated by the teacher, known as Kullback-Leibler (KL) divergence loss (Kullback and Leibler, 1951). In the context of direct ST, KD has been applied not only for model compression, as in its original purpose, but also to enhance the quality of an ST student model by transferring knowledge from an MT teacher capable of achieving superior performance, as demonstrated by Liu et al. (2019).

During my PhD journey, I mostly adopted the multitask learning technique since it is easier to apply and less computationally expensive. In particular, I leveraged the CTC loss for multitask learning in all the selected contributions discussed in this thesis (Sections 3.2.1, 3.2.2, 3.2.3, and 4.2.1). Model pre-training, on the other hand, was only employed in Section 4.2.2, where pre-training was executed on the ST task and then followed by training on the automatic subtitling task.

2.4 Evaluation

The evaluation of standard ST systems is a critical aspect that mainly gauges the quality of the generated translation. However, in the constrained scenarios investigated in this PhD, additional evaluation aspects come into play, as already outlined in Section 1.2.

Evaluation metrics can be categorized into three main groups based on the specific aspect they assess, in our case *quality*, *latency*, and *conformity*. Specifically:

- **Quality**: metrics that quantify the quality of the translation, encompassing accuracy (i.e., how closely the generated translation communicates the original meaning) and fluency (i.e., how smoothly and efficiently the generated translation reads in the target language). Quality metrics are used throughout the whole thesis since this is the primary dimension along which any ST application is evaluated.

- **Latency**: metrics measuring the elapsed time between the spoken utterance and its corresponding translation. Latency metrics are particularly important in the simultaneous ST scenario (Chapter 3), where balancing the trade-off between quality and delay is crucial.

- **Conformity**: metrics assessing the adherence of the output to specific constraints, including:

  - *Length*: metrics evaluating output conformity to length constraints by considering the number of characters or words it comprises;
  - *Duration*: metrics evaluating output conformity in terms of its display duration.

  Conformity metrics are particularly important in the automatic subtitling scenario (Chapter 4), where translation quality should be guaranteed also respecting spatio-temporal constraints focused on user experience.

In assessing the translation quality of ST systems, various metrics have been proposed over time. Traditional metrics, relying on string matching (either at character, n-gram, or word level), include widely adopted measures such as Bilingual Evaluation Understudy or BLEU (Papineni et al., 2002) (usually computed using the sacreBLEU tool (Post, 2018)), character n-gram F-score or chrF (Popović, 2015), Metric for Evaluation of Translation with Explicit ORdering or METEOR (Banerjee and Lavie, 2005), Recall-Oriented Understudy for Gisting Evaluation or ROUGE (Lin, 2004), and Translation

Edit Rate or TER (Snover et al., 2006). More recently, there has been a shift towards embracing metrics that better capture the nuances of spoken language (Freitag et al., 2022) by leveraging, for example, the similarity of sentence or word embeddings (e.g., the embeddings obtained by the BERT model (Devlin et al., 2019)). Among them, COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020), and BERTScore (Zhang* et al., 2020) are the most widespread. Despite the shift, I opted to report BLEU scores in my research, given its common usage in prior works, and, in turn, to enable fair comparisons.

In the context of simultaneous ST, an additional metric measuring latency is required to assess the ability of the system to provide timely aligned translations. Section 3.1.2 will elaborate on the latency metrics commonly adopted for this task and on the new proposals emerging from my PhD work as contributions to this ever-evolving research area.

Within the context of automatic subtitling, both length and duration conformity measures are required for assessing the impact of the subtitles on the screen, alongside translation quality. Therefore, in Section 4.1.2, I provide the reader with information about subtitling guidelines and how adherence to these aspects is evaluated.

Chapter 3

Simultaneous Speech Translation

3.1 Background

Simultaneous speech translation (or 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 for achieving low latency in scenarios, such as streaming conferences and lectures, where the text has to be displayed following as much as possible the pace of the speech. In other words, as the speaker talks, the translated text should appear on the screen as quickly as possible, allowing audiences to understand and engage with the content in real-time. This distinctive requirement sets SimulST apart from traditional offline ASR and ST tasks, which typically process complete utterances before generating the output.

Despite the growing demand,[1] the problem is still far from being solved. SimulST is indeed a very complex task in which the difficulties of performing speech recognition from partial inputs are exacerbated by the problem of projecting meaning across languages. In fact, translating spoken content involves not only understanding utterances in the source language but also conveying that meaning accurately and fluently in the target language. Moreover, this has to be realized while adhering to latency constraints to ensure that the translated text is displayed in real time, adding another layer of complexity as the system has to continuously balance the trade-off between latency and translation quality.

Similar to the history of offline speech translation (Section 2.1), the adoption of cascade architectures was the first attempt made by the SimulST community to tackle

[1] Speech-to-text translation market is expected to grow from USD 2.4 billion in 2022 to USD 5.8 billion in 2027 (`https://www.globalmarketestimates.com/market-report/speech-to-text-market-3839`).

the problem of generating text from incremental input. The first cascade system for SimulST was proposed in 2009 (Fügen, 2009) and was followed by many subsequent studies (Fujita et al., 2013; Niehues et al., 2018b; Xiong et al., 2019; Arivazhagan et al., 2020; Bahar et al., 2021; Iranzo-Sánchez et al., 2022). The cascade paradigm involves a pipeline of two components, in which a streaming automatic speech recognition (ASR) module transcribes the input speech into the corresponding text (Wang et al., 2020c; Moritz et al., 2020), and 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., 2019a; Arivazhagan et al., 2019). This approach, which has been the main solution until 2020, has several intrinsic limitations. First, it suffers from *error propagation* (Sperber and Paulik, 2020), a well-known problem even in the offline scenario (Section 2.1), where the transcription errors made by the ASR module are propagated to the MT module, which cannot recover them as it does not have direct access to the audio. In the SimulST context, this is further complicated by the need to transcribe partial and incremental audio inputs with low latency, potentially obtaining transcriptions of lower quality when accessing only a limited context to generate the output. Another strong limitation of cascaded systems is the *extra latency* added to accomplish the two-step pipeline since the MT module has to wait until the streaming ASR output is produced.

To overcome these issues, the direct model for SimulST proposed by Ren et al. (2020) emerged as a valid alternative to cascade architectures. In proposing the first direct architecture for SimulST, the authors proved that the direct approach is especially promising for SimulST, as it not only avoids the error propagation problem but also reduces the overall latency of the system due to the absence of intermediate symbolic representation steps. These encouraging results have driven recent efforts towards the development of increasingly powerful and efficient models. In fact, despite the data scarcity issue caused by the limited availability of ST corpora (Section 2.3), the adoption of direct architectures for SimulST has gained increasing traction, with a growing number of participants in the IWSLT SimulST Evaluation campaigns[2] (Anastasopoulos et al., 2021, 2022; Agarwal et al., 2023) opting for direct models (Wang et al., 2022; Fukuda et al., 2022, 2023; Gaido et al., 2022e; Huang et al., 2023; Yan et al., 2023; Papi et al., 2023b; Polák et al., 2022, 2023).

Among all aspects of SimulST, two assume a crucial role, motivating their choice as main points for investigation in SimulST research. The first is the **decision policy**, the technique used for deciding when and what to emit to best balance the quality-latency

[2] `https://iwslt.org/`

trade-off, which constitutes one of the basic elements in the development of SimulST architectures. The second is the **evaluation**, the methods and the assumptions used to establish the behaviour and assess the quality and latency performance of SimulST systems in real time. In the next section, the fundamentals of the two aspects will be introduced, followed by an in-depth discussion of the research conducted during the course of my doctoral studies.

3.1.1 Decision Policy

Output generation by a SimulST system is guided by the so-called *decision policy* that is the strategy to decide, at each time step, whether the available information is enough to produce a partial translation, i.e., to perform a *write* action using the audio received until that step, or if it we need to wait and perform a *read* action to receive additional information from the input. Different decision policies result in different ways to balance the quality/latency trade-off. On one side, more read actions will provide the system with a larger context useful to generate translations of higher quality. On the other side, the inherent lagging introduced by each *read* operation will increase, sometimes up to an unacceptable latency.[3]

To address this problem, two types of policy have been proposed: *fixed* and *adaptive*. While fixed decision policies make decisions based on fixed units, such as the number of words present in a speech chunk, adaptive policies make decisions based on contextual information extracted from the input. Fixed policies, tied to predetermined units, offer more precise control over output latency but run the risk of being less accurate, as they may compel the model to generate output even when sufficient context is lacking.

In the following, fixed and adaptive decision policies are described in detail, along with their most representative applications in the literature.

FIXED DECISION POLICY FOR SIMULST

One of the first and the most widely used policies is a **fixed policy** called *wait-k* (Ma et al., 2019a). Simple yet effective, it is based on waiting for k source words before starting to generate the target sentence, and then alternating read and write actions, as shown in Table 3.1.

[3]For instance, the IWSLT 2021 and 2022 SimulST shared tasks define three latency regimes (Anastasopoulos et al., 2021, 2022) – $1s$, $2s$, and $4s$ – while, in IWSLT 2023, a unique threshold is set at $2s$ (Agarwal et al., 2023). Moreover, limits of acceptability have been set between $2s$ and $6s$ for the *ear-voice span* depending on different conditions and language pairs (Yagi, 2000; Chmiel et al., 2017).

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

Table 3.1: wait-k policy example with $k = \{3, 5\}$

As the original wait-k implementation was designed to operate 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. Many subsequent studies have also adopted the wait-k policy with this formulation (Han et al., 2020; Karakanta et al., 2021b; Nguyen et al., 2021; Gaido et al., 2022e; Wang et al., 2022; Fukuda et al., 2022; Liu et al., 2022).

In (Ren et al., 2020), the adaptation to the audio domain 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. Inspired by previous work indicating that emitting fixed chunks of words enhances translation quality with negligible impact on latency (Nguyen et al., 2021), Zeng et al. (2021, 2022) adapted the CTC-based segmentation method to emit chunks of words, allowing re-ranking during the decoding phase. This policy, known as *wait-k-stride-N*, if forced to emit more than one word at a time, N in this case, slightly increasing the latency since the output is prompted after the entire stride is processed (i.e., after the N-th word). This small increase in latency, however, allows the model to perform beam search on the stride, which has been shown effective in improving translation quality (Sutskever et al., 2014).

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. (2020), 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 advantages is that no k value has to be specified for the training, which allows for avoiding the training from scratch of several models for different values of k.

Nevertheless, despite the simplicity in implementing these approaches, fixed policies suffer from a major limitation since they do not consider various aspects of human speech, such as different speech rates, duration, pauses, and silences (Zheng et al., 2020). For this reason, subsequent research efforts on SimulST focused on **adaptive policies** that are able to exploit the information received from the incremental audio input to make a decision.

ADAPTIVE DECISION POLICY FOR SIMULST

After the introduction of fixed policies, several strategies have been developed to directly learn the best policy during training by means of ad-hoc architectures and training procedures aimed at reducing latency with an eye to a more flexible and informed use of contextual information. This category of adaptive policies represents adaptive policies, which can be divided into 3 main groups.

Policies modifying the attention mechanism. This group of policies focuses on the alteration of the attention mechanism present in the ST models without radical interventions in the architecture. The first adaptive policy of this category was proposed in (Ma et al., 2020b), where the Monotonic Multihead Attention or MMA (Ma et al., 2019b) initially introduced for SimulMT was adapted for SimulST. The MMA extends the monotonic attention mechanism (Raffel et al., 2017), which constraints the attention to a fixed reading window represented by previous encoder states, to multihead attention. Later, Ma et al. (2021b) proposed a direct Transformer-based model equipped with an augmented memory Transformer encoder, which augments the attention mechanism with memory banks as originally proposed for streaming ASR with hybrid and transducer-based models. In (Chang and Lee, 2022), the authors proposed an adaptation for SimulST of the Continuous Integrate and-Fire (Dong and Xu, 2020), a variant of the monotonic attention, achieving better results compared to wait-k and MMA. In a similar fashion, Zhang and Feng (2022) adapted the concept of optimal information transport (Villani et al., 2009) to source to target translations, realizing the Information-Transport-based Simultaneous Translation (ITST). ITST quantifies the transported information weight from each source representation to the current target token and injects this information into the attention mechanism so as to decide whether to translate the target token according to its accumulated received audio information.

Policies modifying the architecture. This group of policies focuses on the partial or entire modification of standard encoder-decoder ST architectures. Among these works, one of the most relevant is presented in (Liu et al., 2021a,b), where the Cross Attention

Augmented Transducer (CAAT) model was proposed establishing the SimulST state of the art at that time. CAAT is based on Transformer-Transducers (T-T), originally introduced for ASR (Yeh et al., 2019), which replace the decoder of the standard Transformer encoder-decoder architecture with two components: a *predictor* or label encoder that is independent of the encoder and is in charge of generating a sequence of label embeddings conditioned only on the previously predicted labels, and a *joiner* or joint network that takes the output of both the encoder and the predictor and combines them to compute a distribution over the next label in the output sequence. CAAT modifies the original T-T architecture by adding the cross-attention mechanism to the joiner and is trained also by adopting a latency loss optimized by a forward-backward algorithm. Successively, an improved version of CAAT called Dynamic-CAAT was realized by training across multiple values of the right context window size, achieving good online performance without setting a prior right context window size during training (Zhu et al., 2022). In (Deng et al., 2022), instead, the authors proposed the use of the blockwise Transformer (Tsunoo et al., 2021) for SimulST, achieving better results compared to the wait-k policy. In (Xue et al., 2022), the T-T with its original formulation was proposed as a backbone for SimulST and later extended to the multilingual scenario by (Wang et al., 2023a). Recently, Raffel and Chen (2023) introduced the Implicit Memory Transformer that implicitly retains memory through a new left context method. Following this new method, the left context is computed from the attention output of the previous segment and included in the keys and values of the current segment attention calculation, removing the need to explicitly represent memory with memory banks. Following up the work on Transformer-augmented transducers (CAAT), Tang et al. (2023) proposed a solution by combining Transducer and Attention-based Encoder-Decoder (TAED) that share the same speech encoder, the predictor in the transducer is replaced by the attention-based decoder and the outputs of the decoder are also conditioned on the speech inputs instead of only by the outputs from an unconditioned language model, as in the original transducers. With this new but complicated architecture, the authors achieve new state-of-the-art results in terms of translation quality while being competitive in terms of latency.

Other policies. This category of policies encompasses those that neither directly involve the attention mechanism nor the architecture. In Indurthi et al. (2022), the authors proposed the use of an *external language model* to improve the decision of an MMA-based model. In (Liu et al., 2020b), a unidirectional model was *fine-tuned on partial inputs* to simulate the test conditions (Niehues et al., 2018b), and a policy named Local Agreement, where the agreeing prefixes of two consecutive audio chunks

are considered stable hypotheses and emitted, was used for the inference phase. In (Zaidi et al., 2022), the authors focused on the *training procedure* and introduced the Cross-Modal Decision Regularization (CMDR) loss to improve the learned MMA-based decision policy by computing the similarity between the monotonic attention of speech and text inputs corresponding to each training example. In (Omachi et al., 2023), the authors also worked on the training procedure but, in this case, the SimulST model is trained to generate the unordered output and then reordered later, at the cost, however, of increased latency. This latter method builds upon the interpreter approach, similarly to (Dong et al., 2022) where the policy *learns to segment the source speech* into meaningful units by considering both acoustic features and translation history, maintaining consistency between the segmentation and translation. In the same research line, Zhang et al. (2022a) worked on the *audio segmentation* for SimulST proposing the integrate-and-firing method to learn when to translate the received utterance. More recently, Zhang and Feng (2023) proposed to directly learn segmentation from the underlying translation model. The idea is to turn hard segmentation into differentiable during training, enabling it to be jointly trained with the translation model and thereby learn a segmentation that is more beneficial for translation.

All these methods can be categorized as *learned* adaptive policies since the ST system is – directly or indirectly – adapted for the simultaneous scenario through ad-hoc and often complicated architectures and training procedures. With all these studies on learned adaptive policies in mind, what I identified as an important research question during my PhD studies on SimulST was: **Is adapting these models to the SimulST task necessary? Can we exploit the knowledge already acquired through standard training procedures to guide the ST model during the simultaneous inference? In other words, is there an *intrinsic* adaptive policy within pre-existing ST models?**

Before delving into the answer to these questions (Section 3.2), in the following, I will introduce an essential lens through which to interpret the results: the evaluation metrics.

3.1.2 Evaluation

A good simultaneous model should produce a high-quality translation with reasonable timing, as waiting too long will negatively impact the streaming user experience. The offline MT and ST communities commonly use the firm-grounded BLEU metric (Papineni et al., 2002; Post, 2018) to measure the **quality** of the output translation, but a

simultaneous system also necessitates a metric that accounts for the time spent by the system to output the partial translation, namely a **latency** metric. Given the temporal nature of this metric, lower latency serves as an indicator of superior performance, meaning a reduced delay between speech events and the generation of the corresponding translation. For instance, when one system achieves lower latency compared to another, we refer to this as an improvement in latency for the first system (Figure 3.1).

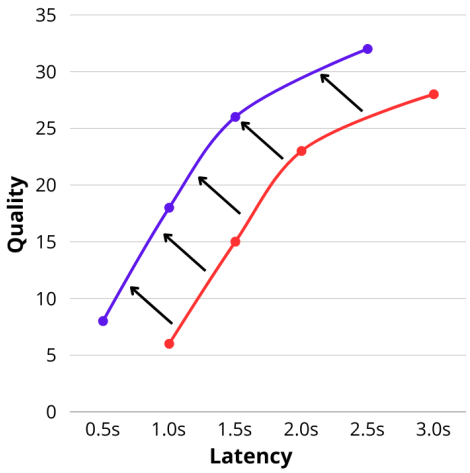


Figure 3.1: Example of an improvement of the simultaneous performance curves: a leftward shift means latency reduction and an upward shift denotes an increase in translation quality.

Since simultaneous MT (SimulMT) was the first yet easiest simultaneous scenario studied by the community, as the cascade was the first approach adopted, a set of metrics was previously introduced for the textual input-output translation part, and later extended to deal with speech inputs.

In the following, I therefore start with an overview of the metrics proposed for SimulMT, which is followed by a discussion on their adaptation to SimulST, and a review of with new metrics specifically proposed for this task.

LATENCY METRICS FOR SIMULMT

The first metric for SimulMT, 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:

$$AP = \frac{1}{|\mathbf{X}||\mathbf{Y}|} \sum_{i=1}^{\mathbf{Y}} d_i \tag{3.1}$$

where $\mathbf{X} = [x_1, .., x_{|\mathbf{X}|}]$ represents the source tokens and $\mathbf{Y} = [y_1, .., y_{|\mathbf{Y}|}]$ represents the predicted translation tokens and delay d_i is defined as the number of tokens read $\mathbf{X}_{1:j} = [x_1, ..., x_j], j < |\mathbf{X}|$ when generating y_i. A major limitation of AP is that this metric is not length invariant, i.e., its value depends on the input and output lengths and is not evenly distributed on the [0, 1] interval. Specifically, values below 0.5 represent models that have lower latency than an ideal policy (which is perfectly synchronous with the received input), and an improvement of 0.1 from 0.7 to 0.6 is much harder to obtain than the same absolute improvement from 0.9 to 0.8 (Ma et al., 2019a), making this metric strongly unreliable.

To overcome these problems, Ma et al. (2019a) introduced *Average Lagging* (AL), which is computed as follows:

$$AL = \frac{1}{\tau(|\mathbf{X}|)} \sum_{i=1}^{\tau(|\mathbf{X}|)} d_i - \frac{i-1}{\gamma} \tag{3.2}$$

where $\gamma = |\mathbf{Y}|/|\mathbf{X}|$, the term $\frac{i-1}{\gamma}$ represents an ideal policy (wait-0) to compare with, and $\tau(|\mathbf{X}|) = \min\{i|d_i = |\mathbf{X}|\}$ is the index of the target token when the policy first reaches the end of the source sentence. The AL value directly describes the lagging behind the ideal policy but, as a downside, it is not differentiable, which is, instead, a useful property, especially if the metric is likely to be added to the system loss computation. For this reason, Cherry and Foster (2019) proposed the *Differentiable Average Lagging* (DAL) introducing a minimum delay of $1/\gamma$ after each operation. The Equation 3.2 becomes:

$$DAL = \frac{1}{|\mathbf{Y}|} \sum_{i=1}^{|\mathbf{Y}|} d'_i - \frac{i-1}{\gamma} \tag{3.3}$$

where

$$d'_i = \begin{cases} d_i, & i = 0 \\ \max(d_i, d_{i-1} + \gamma), & i > 0 \end{cases}$$

LATENCY METRICS FOR SIMULST

The most popular metric, Average Lagging, was 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). Specifically, in SimulST, the input sequence is represented as a stream of audio speech in the source language $\mathbf{X} = [x_1, ..., x_{|\mathbf{X}|}]$ where each element x_j is a raw audio segment of duration T_j,

the reference as a stream of words in the target language $\mathbf{Y}^* = [y_1^*, ..., y_{|\mathbf{Y}^*|}^*]$, and the model translation as a stream of predicted words $\mathbf{Y} = [y_1, ..., y_{|\mathbf{Y}|}]$. In the simultaneous setting, a system starts to generate a partial hypothesis while it continues to receive an incremental stream of input. This implies that, to generate the y_i target word at time j, it has access to $\mathbf{X}_{1:j} = [x_1, ..., x_j]$ with $j < |\mathbf{X}|$.

Therefore, the delay with which the y_i word is emitted is $d_i = \sum_{i=1}^{j} T_i$. Using this notation, in (Ma et al., 2020a), AL was initially defined as follows:

$$AL = \frac{1}{\tau'(|\mathbf{X}|)} \sum_{i=1}^{\tau'(|\mathbf{X}|)} d_i - d_i^* \tag{3.4}$$

$$d_i^* = (i-1) \cdot \frac{\sum_{j=1}^{|\mathbf{X}|} T_j}{|\mathbf{Y}|} \tag{3.5}$$

where $\tau'(|\mathbf{X}|) = \min\{i | d_i = \sum_{j=1}^{|\mathbf{X}|} T_j\}$ is the index of the target token when the end of the source sentence is reached and d_i^* represents an oracle that, perfectly in sync with the speaker, starts to emit words as soon as the speech starts.

However, the authors noticed that this adaptation was not robust for models that tend to stop generating the hypothesis too early or, in other words, that under-generate. This phenomenon is more likely to happen in SimulST than in SimulMT, for which AL was first proposed. For instance, the presence of long pauses in the speech may induce systems to generate the end-of-sentence token too early, even if the source utterance is not yet complete. As observed by the authors, when this phenomenon occurs, the lagging behind the oracle becomes negative. It follows that relatively good latency-quality trade-offs can be achieved thanks to inappropriate AL discounts in case of under-generation, while this does not reflect the reality. Thus, in (Ma et al., 2020a), Equation 3.4 was redefined as:

$$d_i^* = (i-1) \cdot \frac{\sum_{j=1}^{|\mathbf{X}|} T_j}{|\mathbf{Y}^*|} \tag{3.6}$$

assuming that the oracle delays d_i^* are computed based on the reference length rather than on the system hypothesis length.

In a successive work done during this PhD, I pointed out a major issue of AL that arises in opposite conditions, that is in the presence of over-generation. In this case, AL improperly favors over-generating systems, which produce translations that are longer than the reference. In (Papi et al., 2022a), I highlighted the problem through the analysis of empirical outputs generated from existing SimulST systems and, to overcome

this problem, I proposed a new version of the metric called Length-Adaptive Average Lagging (LAAL). To take into account over-generation and allow for fair SimulST systems comparisons, its new formulation modifies the delay definition as follows:

$$d_i^* = (i-1) \cdot \frac{\sum_{j=1}^{|\mathbf{X}|} T_j}{\max\{|\mathbf{Y}|, |\mathbf{Y}^*|\}} \tag{3.7}$$

In this way, neither under-generation nor over-generation is rewarded since, in the case of under-generation ($|\mathbf{Y}| \leq |\mathbf{Y}^*|$), we take the reference length ($|\mathbf{Y}^*|$) while, in the case of over-generation ($|\mathbf{Y}| > |\mathbf{Y}^*|$), we take prediction length ($|\mathbf{Y}|$), thus never summing negative delays.[4]

More recently, another metric was proposed named Average Token Delay (ATD) that focuses on the end timings of partial translations. ATD is formulated as:

$$ATD = \frac{1}{|\mathbf{Y}|} \sum_{i=1}^{|\mathbf{Y}|} (y_i - x_{a(i)}) \tag{3.8}$$

where

$$a(i) = \begin{cases} s(i), & s(i) \leq L_{acc}(x^{c(i)}) \\ L_{acc}(x^{c(i)}), & \text{otherwise} \end{cases} \tag{3.9}$$

$$s(i) = i - \max\left(L_{acc}(y^{c(i)} - 1) - L_{acc}(x^{c(i)-1}), 0\right) \tag{3.10}$$

$T(.)$ in Equation 3.8 represents the ending time of each token, $a(i)$ represents the index of the input token corresponding to y_i, $L_{acc}(x^c) = \sum_{j=1}^{c} |x^j|$ is the cumulative length up to the c-th chunk, and $L_{acc}(x^0) = 0$. $L_{acc}(y^c)$ is defined similarly. $c(i)$ denotes the chunk number c to which y_i belongs. As per Equation 3.10, if the previous translation prefix is longer than the previous input prefix, $s(i)$ becomes smaller than the output index i, which means the previous long output makes the time difference between the input token and the corresponding output token larger. ATD is the average delay of output sub-segments against their corresponding input sub-segments, considering the latency required for inputs and outputs. Although the input-output correspondence does not necessarily mean semantic equivalence, especially for language pairs with large differences in their word order and the numbers of tokens, the authors used this simplified formulation for the latency measurement, the same as AL.

[4]Since its introduction, LAAL has been adopted in the IWSLT Evaluation Campaign on Simultaneous Translation (`https://iwslt.org/2023/simultaneous`).

In parallel, Ma et al. (2020b) raised the issue of using computational unaware metrics, which disregard the actual computational time spent by the model to generate the output. To overcome this issue, they proposed computational aware metrics accounting for the time spent by the model to generate the output. Unfortunately, computing these metrics is not easy in the absence of a unique and reproducible environment that can be used to evaluate the performance of the SimulST models. To this end, in another work (Ma et al., 2020a), they proposed a tool, *SimulEval*, for the metrics computation by simulating a real-time scenario with a client-server 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. This tool will be employed in all the experiments performed during this PhD, where the performance of SimulST, initially assessed in terms of AL, is later evaluated, in the latest works, in terms of LAAL.

3.2 Selected Contributions

In the context of SimulST, almost all the developed systems surveyed in Section 3.1.1 are trained in a simultaneous fashion, i.e., they are trained to simulate the test-time conditions of processing partial incremental input. Since the size of the partial input – and consequently of the context that the SimulST system can use for translation – varies according to the latency requirements imposed by real-world applications, several models are usually trained and maintained to accommodate different quality-latency trade-offs. This applies to both the more simplistic fixed policies presented in Section 3.1.1.1 and the diverse learned adaptive policies overviewed in Section 3.1.1.2.

But is this adaptation to the SimulST task necessary? What if we use an offline-trained ST model for the simultaneous inference? When I started my PhD, the benefits of training SimulST systems on partial inputs were taken for granted and, although works employing offline ST models for SimulST were documented in literature (Nguyen et al., 2021), the indispensability of simultaneous training had never been demonstrated. To fill this gap, I first focused my studies on the systematic comparison between models trained in simultaneous and offline fashion to discover the performance differences in terms of both quality and latency, if any.

This comparison was conducted by applying the popular wait-k fixed policy, using a fixed duration for each word or the CTC predictions to detect the number of words in the audio speech. Additionally, the analysis was extended to diverse architectures (i.e.,

Transformer and Conformer) and data conditions (i.e., with and without sequence-level knowledge distillation to augment the training data). In the case of the simultaneous-trained model, test-time conditions were simulated during training, and the system was optimized to wait for a predefined number of words before starting the output generation (e.g., k was set to different values). Consequently, several models were created, each corresponding to a specific value of k: the larger the k, the higher the latency. In the case of the offline-trained model, the simultaneous policy was applied only at inference time, without any retraining or adaptation for the SimulST task. This means that only one model was trained (or re-used from the offline ST task).

Surprisingly, this analysis revealed that the offline system can achieve, at the same time, competitive or even superior quality and lower latency compared to the simultaneous counterpart. In particular, the comparison of our offline-trained model to the state-of-the-art CAAT model, as introduced in Section 3.1.1.2, exposed that the offline-trained model could even match or surpass the CAAT performance in the medium-high latency regime (i.e., with computationally aware $AL > 1.5s$), despite neither applying complicated training procedures nor using complex architectures as the CAAT is. Given the positive results achieved by offline-trained models used in simultaneous, my first selected contribution (**PAPER #1**: "*Does Simultaneous Speech Translation need Simultaneous Models?*", Section 3.2.1) represented the basis of my subsequent research on leveraging offline ST models for SimulST.

Having confirmed the efficacy of employing offline-trained models for SimulST, as anticipated in Section 3.1.1, the subsequent research questions on which I focused during my doctoral studies are: **Can we exploit the knowledge already acquired through standard training procedures to guide the ST model during the simultaneous inference? In other words, is there an *intrinsic* adaptive policy within pre-existing ST models?** Therefore, the next step was to identify the mechanisms within the already existing offline ST models that could be exploited to "guide" the model during the simultaneous inference. To this end, I focused on the attention mechanism and, in particular, the cross-attention mechanism responsible for capturing the relationships between audio input and textual output in an ST system. I analyzed the behavior of the model by varying the decoder layer as well as the attention head from which to extract the attention scores.

I observed that the ST system effectively represents alignments between audio and textual representations, a behavior consistent with findings in other tasks like MT (Tang et al., 2018; Zenkel et al., 2019; Garg et al., 2019; Chen et al., 2020). Additionally, I found that the later decoder layers were more representative than the initial ones,

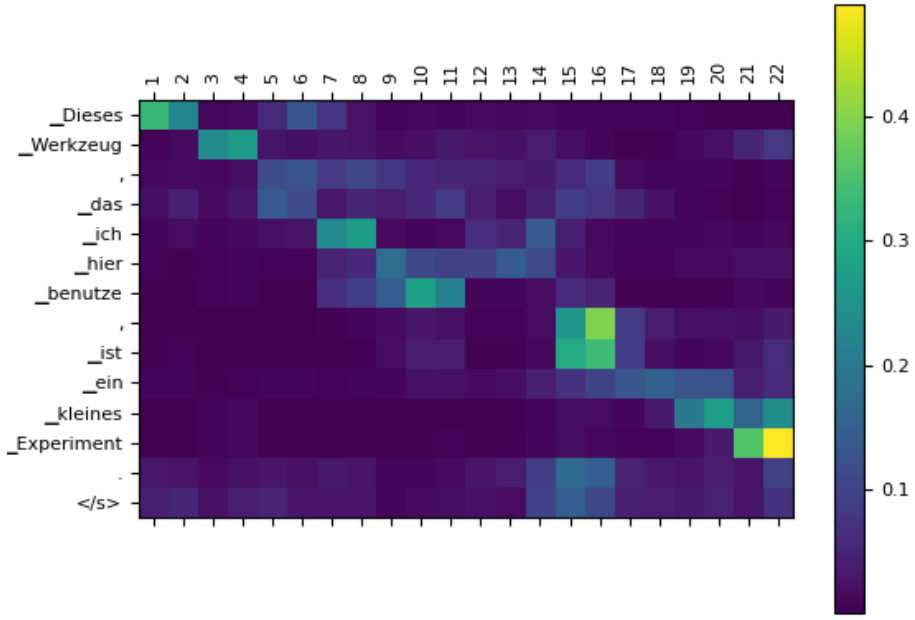


Figure 3.2: Example of an audio-text alignment extracted from cross attention.

and the best representation was achieved by averaging information across the attention heads. An example of audio-text alignment is presented in Figure 3.2.

In light of these findings, I introduced a new adaptive policy called EDAtt, which exploits the cross-attention scores to determine when to emit a partial hypothesis. The approach involves summing the attention scores towards the last λ frames of each token and checking if this sum does not exceed a threshold α. The underlying hypothesis is that, if this is true, the received encoded information can be considered stable enough to emit the token; otherwise, the emission is stopped and the system waits for the next speech chunk. The rationale behind EDAtt is that when attention points to the most recent speech information, indicating higher scores towards the last audio frames received, this information might be incomplete and, therefore, still insufficient to generate the token.

By comparing the EDAtt policy with the CAAT model and two policies, the fixed wait-k (Section 3.1.1.1) and the adaptive Local Agreement (Section 3.1.1.2), also applied to the same offline ST model, EDAtt achieved the best quality-latency trade-off. With the only exception of the very low latency scenario and the computational-unaware latency metric ($AL \leq 1s$), EDAtt resulted in the best model for all latency regimes both computationally and non-computationally aware. Being tested on the same environment (with the CAAT model included, which was replicated for **PAPER #1**), the computational-aware latency comparison between the models was totally fair, and

the obtained results proved that not only EDAtt achieves the lowest latency and the highest quality but also that CAAT model is very slow mostly due to its complex architecture.

With my second selected contribution (**PAPER #2**: "*Attention as a Guide for Simultaneous Speech Translation*", Section 3.2.2), I established that there is no need to influence or adapt the behavior of the attention weights through dedicated training strategies, unlike other works in SimulST (Zaidi et al., 2021, 2022; Chang and Lee, 2022; Zhang and Feng, 2022), and that this intrinsic knowledge acquired by the attention mechanism can be directly exploited for SimulST through a novel policy, `EDAtt`, effectively achieving low latency with minimal computational costs.

Applying EDAtt requires finding the optimal value of λ ($\lambda = 2$ in the paper), while the hyper-parameter α directly controls the latency. This decision is usually taken by evaluating the performance of the model on a dev set, which, however, can be very data-dependent. For this reason, in subsequent research, I aimed to simplify the EDAtt policy formulation to rely on a single hyper-parameter dedicated solely to managing the latency of the SimulST system. Building on this research direction, in my third and most recent contribution (**PAPER #3**: "*AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation*", Section 3.2.3), I proposed an innovative alternative to EDAtt named AlignAtt.

In AlignAtt, the attention weights are used to assign each encoder state, corresponding to specific time frames, with a token in the partial hypothesis. This assignment is realized by selecting the maximum cross-attention score, extracted following the same strategy of EDAtt, for each frame. This way, every frame is uniquely aligned with its corresponding token. At each step, by considering the last f frames as "forbidden frames", the partial hypothesis is emitted until a token is assigned to one of the f forbidden frames. This approach reduces the hyper-parameters that handle the latency of the SimulST system to a single parameter: the number of forbidden frames f. Notably, although the simplified formulation, dedicated experiments involving various target languages demonstrated that AlignAtt can match or even outperform all the other policies applied to offline-trained models, including the EDAtt policy, thus representing the new state of the art.

SUMMARY

All in all, in this PhD journey through simultaneous speech translation, I proposed a paradigm shift in how to develop models for the SimulST task. Rather than relying

on purpose-built models exclusively designed and trained for SimulST, I advocated for leveraging offline-trained ST models that may already exist, obviating the need for retraining, modification, or adaptation to suit the simultaneous task. Remarkably, this transition unfolded seamlessly in terms of performance, with both latency and translation quality remaining almost uncompromised.

With my first contribution, I was able to attain highly competitive performance with the state of the art. As my journey progressed, the primary objective evolved into the development of more sophisticated policies while retaining the commitment to enhancing results without the need for resource-intensive, time-consuming training.

This commitment culminated in the development and introduction of two innovative policies: EDAtt and, later, AlignAtt. These policies serve as proof of their simplicity in implementation and their suitability for offline-trained ST models while consistently maintaining the best balance between quality and latency.

In the subsequent sections (Sections 3.2.1, 3.2.2, and 3.2.3), these three major contributions are elaborated upon, as they best represent my journey through the SimulST scenario, and can be summarized as:

- **PAPER #1 (Papi et al., 2022b):**
  - **Publication details**:
    - ***Title***: Does Simultaneous Speech Translation Need Simultaneous Models?
    - ***Authors***: Sara Papi, Marco Gaido, Matteo Negri, Marco Turchi
    - ***Venue***: Finding EMNLP 2022
  - **Research Question(s)**: *Is the adaptation of architectures and training procedures to the SimulST task necessary? What if we use an offline-trained ST model for the simultaneous inference?*
  - **Main Contribution(s)/Finding(s)**: Offline-trained ST systems can achieve competitive or even superior quality and latency compared to the systems trained in simultaneous.
- **PAPER #2 (Papi et al., 2023d):**
  - **Publication details**:
    - ***Title***: Attention as a Guide for Simultaneous Speech Translation
    - ***Authors***: Sara Papi, Matteo Negri, Marco Turchi

    - ***Venue***: ACL 2023
  - **Research Question(s)**: *Can we exploit the knowledge already acquired through standard training procedures to guide the ST model during the simultaneous inference?*
  - **Main Contribution(s)/Finding(s)**: The knowledge acquired by an offline-trained model and, in particular, the cross-attention information can be directly exploited for SimulST, effectively achieving low latency with minimal computational costs.

- **PAPER #3 (Papi et al., 2023e):**
  - **Publication details**:
    - ***Title***: AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation
    - ***Authors***: Sara Papi, Matteo Negri, Marco Turchi
    - ***Venue***: INTERSPEECH 2023
  - **Research Question(s)**: *Can we further improve the way in which we look at the cross-attention scores to enhance the ST performance in real time?*
  - **Main Contribution(s)/Finding(s)**: Utilizing cross-attention information to extract speech-translation alignment and employ it as guidance for simultaneous inference not only offers a straightforward formulation but also delivers the best trade-off between quality and latency.

3.2.1 PAPER #1

Does Simultaneous Speech Translation Need Simultaneous Models?

INTRODUCTION

Many application contexts, such as conferences and lectures, require automatic speech translation (ST) to be performed in real time. To meet this requirement, Simultaneous ST (SimulST) systems strive not only for high output quality but also for low latency (i.e. the elapsed time between the speaker's utterance of a word and the generation of its translation in the target language). Balancing quality and latency is extremely complex as the two objectives are conflicting: in general, the more a system waits – which implies higher latency – the better it translates thanks to a larger context to rely on.

SimulST models manage the quality-latency trade-off by means of a decision policy: the rule that determines whether a system has to wait for more input or to emit one or more target words. The most popular decision policy is the *wait-k*, a straightforward heuristic that prescribes waiting for a predefined number of words before starting to generate the translation. Initially proposed by Ma et al. (2020b) for simultaneous machine translation (SimulMT), the *wait-k* is now widely adopted in SimulST (Ma et al., 2020b; Ren et al., 2020; Han et al., 2020; Chen et al., 2021; Zeng et al., 2021; Ma et al., 2021b) thanks to its simplicity. Apart from *wait-k*, other attempts have been made to develop decision policies learned by the SimulST system itself (Ma et al., 2019b; Zaidi et al., 2021; Liu et al., 2021a,b), all resulting in computationally expensive models with limited diffusion.

Regardless of the decision policy, SimulST systems are usually trained to simulate the conditions faced at inference time, that is with only a partial input available (Ren et al., 2020; Ma et al., 2020b; Han et al., 2020; Zeng et al., 2021; Ma et al., 2021b; Zaidi et al., 2021; Liu et al., 2021a). Since the size of the partial input – and consequently of the context that the SimulST system can exploit to translate – varies according to the latency requirements imposed by real-world applications,[5] several models must be trained and maintained to accommodate different quality-latency trade-offs. This results in high computational costs that contrast with rising awareness on the need to reduce

[5]For instance, the IWSLT SimulST shared task defines three latency regimes (Anastasopoulos et al., 2021) – $1s$, $2s$, and $4s$ – and limits of acceptability have been set between $2s$ and $6s$ for the *ear-voice span* depending on different conditions and language pairs (Yagi, 2000; Chmiel et al., 2017).

energy consumption (Strubell et al., 2019) towards more sustainable AI (Vinuesa et al., 2020; Schwartz et al., 2020).

So far, the benefits of training systems on partial inputs have been taken for granted and, although works employing models trained in offline mode are documented in literature (Nguyen et al., 2021; Ma et al., 2021b), the indispensability of simultaneous training in SimulST has never been demonstrated. With an eye on the burden and environmental impact of training multiple dedicated models for different tasks – offline, simultaneous – and latency regimes, in this work we address the following question: *Does simultaneous speech translation actually need models trained in simultaneous mode?* To this end, we experiment with a single, easy-to-maintain offline model, which can effectively serve both the simultaneous and offline tasks. Specifically, we explore the application of the widely adopted *wait-k* policy to the offline-trained ST system only at inference time, bypassing any additional training neither to adapt the model to the simultaneous scenario nor to accommodate different latency requirements. Through experiments on two language directions (en→{de, es}), having respectively different and similar word ordering with respect to the source, we show that:

- In terms of sustainability, offline training yields considerable reductions – by a factor of 9 in our evaluation setting – in carbon emission and electricity consumption (Sections 3.2.1.4).
- The offline-trained model outperforms or is on par with those trained in simultaneous within the *wait-k* policy framework (Section 3.2.1.5);
- Recent advancements in offline architectures and training strategies further improve output quality without affecting latency (Section 3.2.1.6);
- The effectiveness of offline training also emerges in comparison with the state of the art in SimulST (Liu et al., 2021b): except for the lowest latency regime, our system is superior in the 2s-4s latency interval (ear-voice span) with gains up to 4.0 BLEU (Section 3.2.1.7).

BACKGROUND

3.2.1.2.1 *wait-k*

The *wait-k* policy requires waiting for a predefined number of words before starting to translate. For instance, a system using a wait-3 policy generates the 1st target word

when it receives the 4th source word, the 2nd target word when it receives the 5th source word, and so on. The number of words to wait is controlled by the k parameter. SimulST systems based on the *wait-k* policy are usually trained considering the same k used for testing (Ren et al., 2020; Ma et al., 2020b; Zeng et al., 2021) while, in theory, its value can be different between the training and testing phases. A parameter k_{train} can indeed be used to mask words at training time, while a parameter k_{test} can be used to directly control the latency of the system at inference time according to the requirements posed by the target application scenario.

Since many values of k_{train} can be used to train the SimulST systems, even for identical values of k_{test}, the standard approach involves performing several trainings to obtain the best translation quality while satisfying different latency requirements. In SimulMT, Elbayad et al. (2020) tried to avoid this large number of experiments by exposing the model to different values of k_{train} sampled at each iteration. Surprisingly, they achieve the best performance on several k_{test} using a single value of k for training ($k_{train} = 7$). However, it is not clear if such a rule applies to SimulST, leaving the problem of performing a large number of trainings still unsolved.

3.2.1.2.2 Word detection for *wait-k* in SimulST

Since SimulMT operates on a stream of words, applying the *wait-k* is straightforward because the number of received words is explicit in the input. Conversely, its application to SimulST is complicated by the fact that the input is an audio stream and the number of received words has to be inferred by means of a so-called *word detection* strategy.

Two main categories of word detection strategies are currently employed by the community: fixed (Ma et al., 2020b), and adaptive (Ma et al., 2020b; Ren et al., 2020; Zeng et al., 2021; Chen et al., 2021). The fixed strategy is the easiest approach, as it assumes that a fixed amount of time is required to pronounce every word disregarding the information actually contained in the audio. In contrast, adaptive word detection determines the number of uttered words by looking at the content of the audio. This can be done either by means of an Automatic Speech Recognition (ASR) decoder (Chen et al., 2021),[6] or by means of a Connectionist Temporal Classification (Graves et al., 2006) – CTC – module (Ren et al., 2020; Zeng et al., 2021), every time a speech chunk is received by the system.

[6]This solution involves the use of two separate synchronized decoders (one for simultaneous ASR and one for ST) and will not be analyzed in this work due to the higher computational costs of training a double decoder architecture.

In its simplicity, the fixed strategy does not consider various aspects of the input speech, such as different speech rates, duration, pauses, and silences. For instance, if there are no words in the speech (e.g. in the case of pauses or silences), the fixed strategy forces the system to output something even if it cannot rely on sufficient context. In the opposite case, in which more than one word is pronounced in a speech chunk, the fixed strategy forces the emission of only one word, consequently accumulating a delay. By trying to guess the actual number of words contained in a speech chunk, the adaptive strategy is in principle more faithful to these audio phenomena. However, conflicting results are reported in the literature, some in support of the adaptive strategy (Zeng et al., 2021) while others show no advantage from its application (Ma et al., 2020b).

METHOD

While at training time the SimulST system has the entire audio available, at inference time it receives a partial, incremental input. This mismatch between offline training and simultaneous testing makes the system vulnerable to exposure bias (Ranzato et al., 2016). To mitigate this potential problem, SimulST models are trained under simulated simultaneous conditions. On an attentive model, this simultaneous training is realized by masking future audio frames when computing the encoder-decoder attention. For a *wait-k* SimulST system, the choice of the audio frames to be masked depends on two factors: the value of k_{train} and the word detection strategy. The k_{train} value determines the number of source words to mask (e.g., in the case of wait-3, the first target word is generated by looking at the first three source words and so on). The word detection strategy identifies the source words from the audio by detecting the number of frames each one corresponds to. Thus, the encoder-decoder attention is computed by limiting each target word to only attend to the audio frames that correspond to the previous k_{train} source words identified by the word detection strategy. As a result, testing different word detection strategies requires training several systems, which in turn are trained with different values of k_{train} to obtain different latencies.

In this paper, we question the need for all these experiments by investigating whether the simultaneous training of the ST systems is indispensable to obtain a good quality-latency trade-off. Within the framework of the *wait-k* policy, we explore the ability to translate in real-time of an offline-trained system that is neither trained nor adapted to the simultaneous scenario. To obtain a simultaneous prediction from the offline system, we add a *pre-decision module* after the encoder at inference time. Its role is to incorporate the logic of the word detection strategy to decide whether to wait or to

emit words when a new speech chunk is received, according to the selected k_{test}. In particular, it takes as input the encoder states representing the received audio chunk and applies the word detection strategy (either fixed or adaptive) to obtain the number of source words present in the input. If this number is equal to or exceeds k_{test}, the module activates the decoding part of the model and a word is emitted, otherwise it keeps reading the source speech.

Since the offline system is not trained for the simultaneous task, the choice of k_{test} and word detection strategy are not constrained to those used during training as in the native SimulST case. Indeed, an offline model is trained by always attending to the entire source input. Different from the simultaneous training mode, the encoder-decoder attention is computed without masking, that is by considering past, current, and future information. Although this avoids multiple training for each k_{train} and word detection strategy, it also exposes the model to operate in conditions different from its training setup, as it is not used to receive partial inputs. To check if the exposure bias given by this mismatch in training and testing conditions constitutes a real limitation, we conduct a systematic analysis of the quality-latency performance of the offline-trained system in the simultaneous scenario. To this aim, we compare the offline-trained system with the same model trained in simultaneous mode by varying the value of k_{train} and the word detection strategy.

EXPERIMENTAL SETTINGS

We perform all our experiments on the en→{de, es} sections of the MuST-C dataset (Cattoni et al., 2021). All the results presented are given on the corpus test set (tst-COMMON). We use the Transformer architecture (Vaswani et al., 2017) with the integration of the CTC in the encoder (Liu et al., 2020c; Gaido et al., 2021a), which is used to realize the adaptive word detection strategy. The hyper-parameters, training and inference details are presented in Appendix 3.2.1.9.1.

For the evaluation, we adopt BLEU[7] (Post, 2018) for quality, and Length Adaptive Average Lagging (Papi et al., 2022a) – or LAAL – for latency, which is the modified version of the popular Average Lagging for speech (Ma et al., 2020b) that correctly evaluates both shorter and longer predictions with respect to the reference. We report the simultaneous results in LAAL-BLEU graphs where each curve corresponds to a system trained using a different value of k_{train} and each point to a different k_{test}. The set of k values used for both training the simultaneous model and testing all the models is

[7]BLEU+case.mixed+smooth.exp+tok.13a+version.1.5.1

$k = \{3, 5, 7, 9, 11\}$. We also report the results of the offline generation using the greedy search and the beam search with the $beam\_size = 5$ commonly used in offline ST.

Carbon Footprint. Each training contributed an estimated 70.3 kg of CO_{2eq} to the atmosphere and used 184.7 kWh of electricity. This assumes 116 hours of runtime, a carbon intensity of 380.539g CO_{2eq} per kWh, 4 NVIDIA Tesla K80 GPUs (utilization 93%), and an Intel Xeon CPU E5-2683 v4 (utilization 100%).[8] This means that training a single offline model instead of a model for each value of k_{train} (in our case, 5 models) and for each word detection strategy (in our case, 2 strategies) allows us to save $5 \cdot 2 - 1 = 9$ experiments, amounting to 632.7 kg of CO_{2eq} and 1662.3 kWh of electricity for each language.

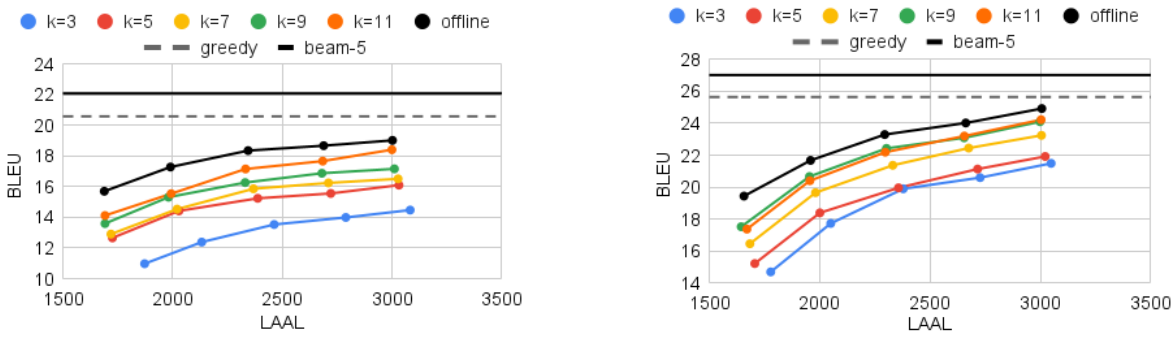
RESULTS

Fixed Word Detection. The results of the *wait-k* models with fixed word detection are shown in Figure 3.3. The LAAL-BLEU curves indicate that the latency of all the systems lies between $1700ms$ and $3000ms$, staying in a medium-high latency regime[9] for both language pairs. Translation quality is lower for en-de, for which it ranges from 11 to 19 BLEU, while for en-es it ranges from 14 to 25 BLEU. The difference in performance between the two language pairs is coherent with the results of the offline generations (both greedy and beam-5) and justified by the different levels of difficulty when translating into the two target languages (having respectively similar and different word ordering with respect to English). The curves of the simultaneous-trained systems also show a tendency: if k_{train} increases, both the quality and latency improve (e.g. on en-de, the k=11 curve lies higher – indicating better quality – and more leftward – lower latency – than the others). Interestingly, the offline-trained models (in solid black) outperform the systems trained in simultaneous at every latency regime, with gains from 1 to 7 BLEU for en-de and from 1 to 6 BLEU for en-es. This indicates that, to achieve the best performance and independently from the k_{test} used, the offline-trained model represents the best choice, at least for the fixed strategy.

Adaptive Word Detection. The results of the *wait-k* models with adaptive word detection are shown in Figure 3.4. The systems latency lies between $1700ms$

[8]The social cost of carbon uses models from (Ricke et al., 2018) and carbon emissions information was estimated using the *experiment-impact-tracker* (Henderson et al., 2020).

[9]Henceforth referring to (Anastasopoulos et al., 2021), we consider three latency regimes depending on the delay d between the time in which the speech is heard and the output translation is received. These are: *low* when d<1000ms, *medium* when $1000 < d < 2000ms$, and *high* when d>2000ms.



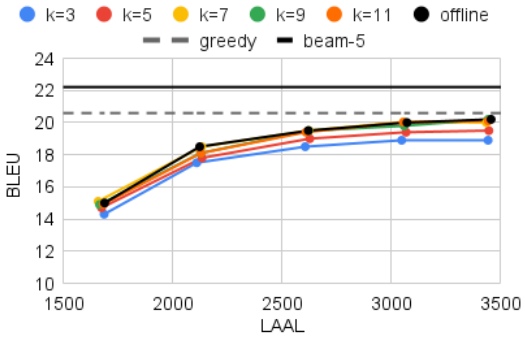
(a) English → German

(b) English → Spanish

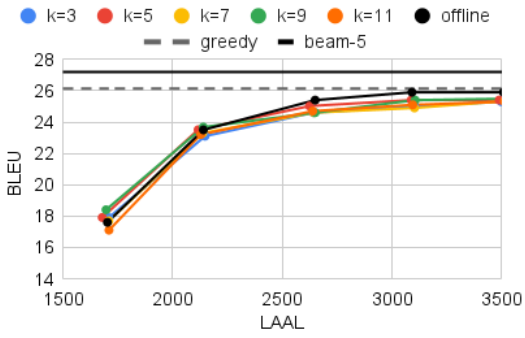
Figure 3.3: LAAL-BLEU curves of *wait-k* with fixed word detection strategy.

and 3500ms and, as with the fixed strategy, the quality is higher for en-es (from 15 to 26 BLEU) than for en-de (from 14 to 20 BLEU). Looking at Figures 3.3 and 3.4, we observe that the overall translation quality yielded by the adaptive strategy is higher compared to that of the fixed one. Moreover, the fixed strategy curves are far from being comparable with their offline greedy values (dashed lines), while the adaptive strategy curves almost reach them at higher latency. However, the models with fixed word detection perform better at lower latency, with a gain of 1 BLEU for en-de and 2 BLEU for en-es. In light of these results, there is not a clear winner between the two word detection strategies. From Figure 3.4, we also notice that the adaptive curves are very close to each other, in contrast with the fixed case. This phenomenon indicates that, in the case of the adaptive strategy, changing k_{train} does not significantly influence the model performance. This suggests that the offline-trained model (comparable to a model trained with $k_{train} = \infty$) should be on par with the simultaneous-trained ones, a consideration corroborated by the trend of the offline-trained system curves (in solid black) that are always above or on par with those of the simultaneous-trained systems.

All in all, we can conclude that, when using the *wait-k* policy, **the offline-trained model achieves similar or even better results compared to the same models trained in simultaneous mode**. Based on this finding, in the next section, we explore the actual potential of offline training for SimulST by adopting the most promising offline architectures and training techniques to improve the quality-latency balancing of our systems.



(a) English → German



(b) English → Spanish

Figure 3.4: LAAL-BLEU curves of *wait-k* with adaptive word detection strategy.

LEVERAGING OFFLINE SOLUTIONS

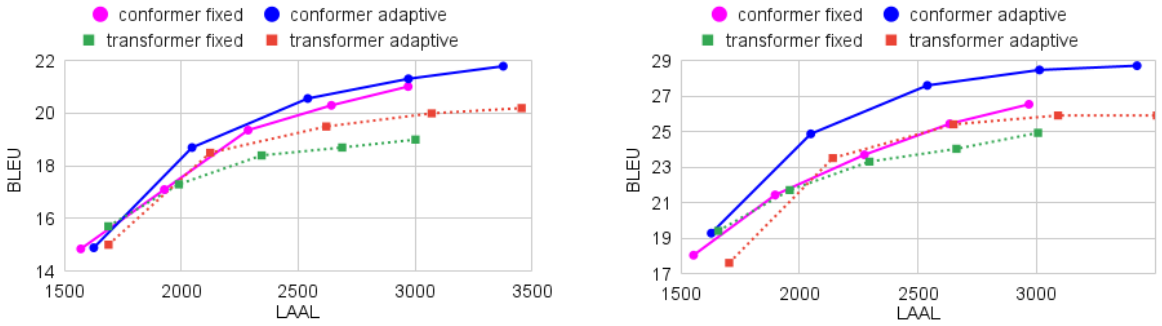
Offline training brings considerable advantages in terms of reducing the computational costs of SimulST technology. First, only one model can be trained and maintained to serve both offline and simultaneous tasks without performance degradation. Second, contrary to the simultaneous training mode, the choice of the word detection strategy at run-time does not depend on the strategy used during training. Rather, it can be made according to the specific use case, making the offline-trained model more flexible. This also means that other decision policies can be applied to the offline-trained system without the need to re-train it from scratch.

Using a single offline-trained model not only speeds up its development but also opens up the possibility to directly adopt powerful offline architectures and techniques without performing any additional training or adaptation to the simultaneous scenario. In the following, we test this hypothesis to find out whether recent architectural improvements (Section 3.2.1.6.1) and data augmentation techniques (Section 3.2.1.6.2) designed for offline ST also have a positive impact on SimulST.

In recent years, many architectures have been proposed to address the offline ST task (Wang et al., 2020a; Inaguma et al., 2020; Le et al., 2020; Papi et al., 2021b). Among them, the Conformer (Gulati et al., 2020) has recently shown impressive results both in speech recognition, for which it was initially proposed, and in speech translation (Inaguma et al., 2021a). The main aspects characterizing this encoder-decoder architecture are related to the encoder part. Inspired by the Macaron-Net (Lu et al., 2019), the Conformer encoder is built with a sandwich structure and integrates the relative sinusoidal positional encoding scheme (Dai et al., 2019).

Given the promising results it achieved in the offline scenario, we choose to test if

this architecture also brings quality and latency gains in SimulST. Since we found in Section 3.2.1.3 that fixed and adaptive word detection strategies have their own use cases (their best results are observed at different latency regimes, respectively low for fixed and medium-high for adaptive), we compare Conformer- and Transformer-based architectures using both strategies. For the offline training of Conformer, we follow the same procedure used for Transformer. Details about the model hyper-parameters are presented in Appendix 3.2.1.9.1.



(a) English → German (b) English → Spanish

Figure 3.5: LAAL-BLEU curves of the Transformer- and Conformer-based architectures.

3.2.1.6.1 Scaling Architecture

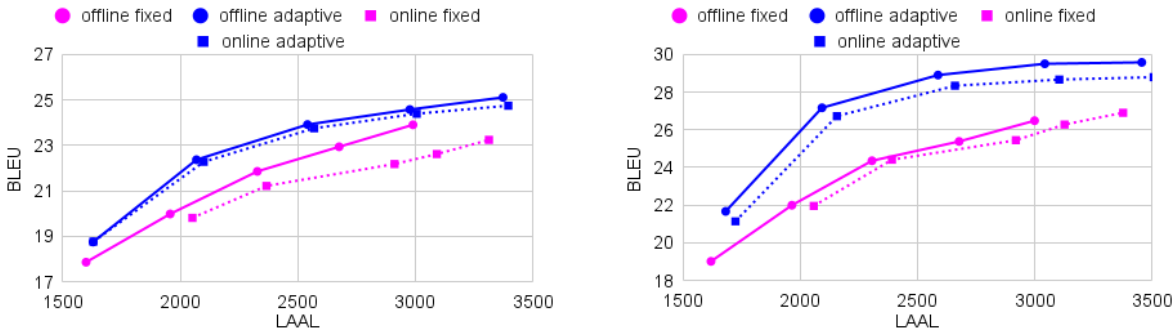
The offline results of both architectures are presented in Table 3.2, while their simultaneous curves are shown in Figure 3.5.

As previously noticed by Inaguma et al. (2021a), Conformer outperforms Transformer in offline generation. The improvements, of at least 2.4 BLEU points, are valid both for greedy and beam search. From Figure 3.5, we can see that Conformer also outperforms Transformer in the simultaneous setting. This holds both for fixed and adaptive word detection, with larger BLEU gains at higher latency regimes. As far as word detection strategies are concerned, we also notice a similar trend between Conformer and Transformer: the fixed one performs better or on par at lower latency while being outperformed by the adaptive one when the latency increases.

In light of the better results obtained by Conformer, we conclude that **improving the architecture of the offline system also has a positive impact on its simultaneous performance**, enhancing translation quality without affecting latency.

Model	En-De		En-Es	
	greedy	beam-5	greedy	beam-5
Transformer	20.6	22.2	26.1	27.2
Conformer	23.3	24.8	28.5	29.6

Table 3.2: BLEU results of the offline generation.



(a) English → German

(b) English → Spanish

Figure 3.6: LAAL-BLEU curves of offline- and simultaneous-trained Conformer models with sequence-level KD.

3.2.1.6.2 Scaling Data

Data augmentation is a common practice used to improve systems performance. One approach to data augmentation is to apply knowledge distillation (KD), which was introduced to transfer knowledge from big to small models (Hinton et al., 2015). Among the possible methods, sequence-level KD (Kim and Rush, 2016) is one of the most popular ones in ST thanks to its application simplicity and the consistent improvements observed (Potapczyk and Przybysz, 2020; Xu et al., 2021; Gaido et al., 2022c). Sequence-level KD consists of replacing the target references of a given parallel training corpus with the predicted sequences generated by a teacher model (usually, an MT model), from which we want to distil the knowledge to a student model.

To investigate the effects of such a knowledge transfer on quality and latency, we apply sequence-level KD to our offline-trained SimulST system. To this end, we translate the transcripts present in the en→{de, es} sections of MuST-C with an MT model (more details are provided in Appendix 3.2.1.9.1) and we substitute the gold translations with the MT-generated ones to build new data. As in (Liu et al., 2021b), to train the models we use both gold and synthetic data by concatenating them. Since the performance of the Conformer model scales with data (Gaido et al., 2022e) and is better compared to that of Transformer (Section 3.2.1.6.1), we adopt the Conformer for the following

study. We extend our analysis to the simultaneous-trained systems to verify if the offline-trained one continues to perform at least on par with them and we report the best simultaneous-trained system curve for each word detection strategy.

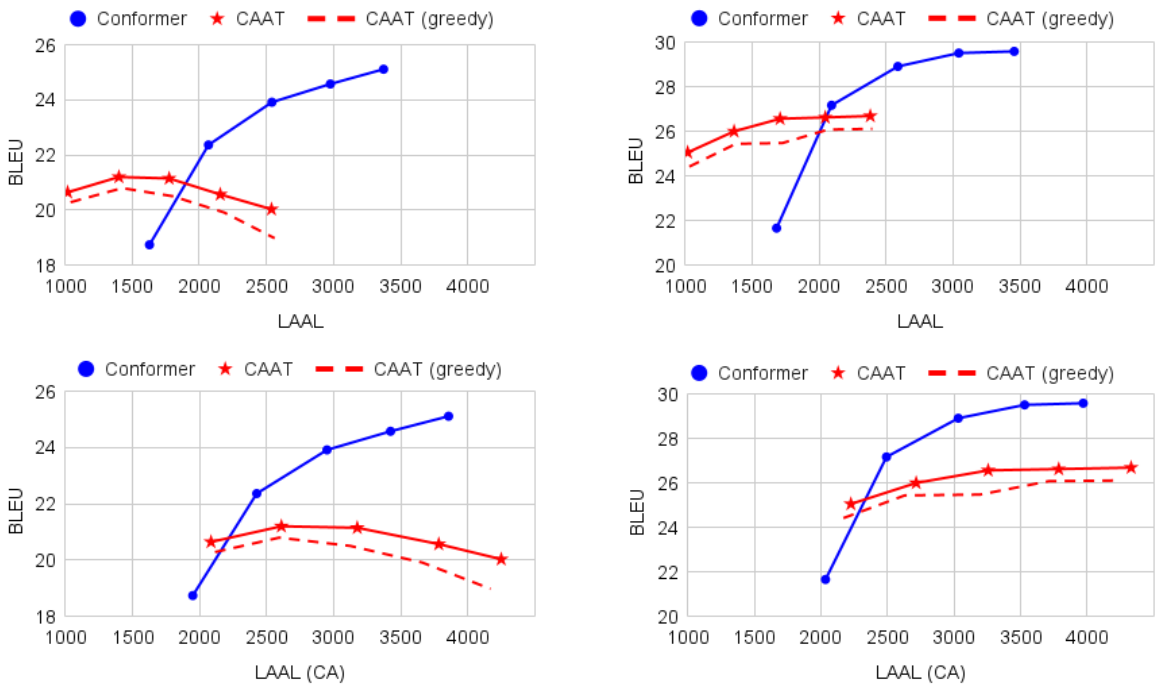
The effects of the additional KD data are shown in Figure 3.6. Compared to Figure 3.5, we notice a performance improvement that comes without sacrificing latency. On en-de, the quality of the offline-trained Conformer with KD ranges from 18 to 25 BLEU, against the previous 15 to 22 BLEU. On en-es, it ranges from 19 to 30 BLEU, against the previous 18 to 29 BLEU. Moreover, the offline-trained system (solid curves) is still better or at least comparable with the simultaneous-trained ones (dotted curves) for both language pairs. From Figure 3.6, we also notice that adaptive word detection (blue curves) shows overall better results compared to the fixed one (pink curves), even at lower latency. This suggests that comparing the two strategies by using models with higher translation quality shows the superiority of adaptive word detection at any latency regime.

In light of these results, we conclude that **data augmentation improves the offline-trained system quality without affecting latency**. To better assess these performance gains in the simultaneous framework, in the next section we present a detailed comparison of our offline-trained Conformer with the state-of-the-art SimulST architecture.

COMPARISON WITH THE STATE OF THE ART

So far, we discovered that scaling to better-performing architectures and more data further improves the simultaneous results of offline-trained models. But how good is their performance compared to the state of the art in SimulST? To answer this question, we compare our best system, the offline-trained Conformer with adaptive word detection, with the Cross Attention Augmented Transducer (Liu et al., 2021b) – CAAT – used by the winning submissions at IWSLT 2021 Anastasopoulos et al. (2021) and 2022 Anastasopoulos et al. (2022). Inspired by the Recurrent Neural Network Transducer by Graves (2012), CAAT is made of three Transformer stacks: the encoder, the predictor, and the joiner. These three elements are jointly trained in simultaneous to optimize the quality of the translations while keeping latency under control.

For training and testing the CAAT architecture, we use the code published by the authors and adopt the same hyper-parameters of their paper. As the performance of the CAAT model is sensitive to sequence-level KD (Liu et al. 2021b show a 2 BLEU degradation without it), we compare it with the offline-trained Conformer model using



(a) English → German

(b) English → Spanish

Figure 3.7: LAAL/LAAL_{CA}-BLEU curves of our offline-trained Conformer and state-of-the-art (CAAT) models.

the same data settings – see Section 3.2.1.6.2. We report the CAAT results obtained by adopting both the greedy search used in our SimulST settings and the beam search used by Liu et al. (2021b). As suggested by Ma et al. (2020b), we also compute the Computational Aware (CA) version of the LAAL metric (LAAL_{CA}), which is defined as the time elapsed from the beginning of the generation process to the prediction of the partial target.[10] Since LAAL_{CA} represents the real wall-clock elapsed time experienced by the user, it gives a more reliable evaluation of the SimulST performance in a real-time scenario. For the sake of completeness, we also report the results of Average Lagging (Ma et al., 2020a) in Appendix 3.2.1.9.3.

We present the comparison in Figure 3.7. From the LAAL-BLEU curves, we see that, at low latency regime, the CAAT model (in solid red) outperforms our offline-trained Conformer model (in solid blue) by 2 BLEU on en-de and 4 BLEU on en-es. However, moving to medium-high latency regime, the Conformer significantly outperforms CAAT,

[10]Given that LAAL_{CA} depends on the computation time, we perform all the generations on one NVIDIA Tesla K-80 GPU and provide the results by averaging over 3 runs. However, we notice a very small variance among the runs (in the order of $10ms$), suggesting that averaging is not necessary to provide sound results.

reaching gains of 4 BLEU on en-de and 2 BLEU on en-es. We can also notice a degradation of the CAAT en-de translation quality that is caused by an under-generation problem at higher latency, for which we give details in Appendix 3.2.1.9.2.

When it comes to LAAL_{CA}-BLEU, the scenario changes, bringing CAAT curves much closer to those of Conformer. The state of the art still outperforms the Conformer at lower latency but in this case, waiting about $100/200ms$ more, the Conformer performance starts to improve consistently.

Comparing the LAAL- and LAAL_{CA}-BLEU curves, we see that our offline-trained system is more coherent between computational and non-computational aware metrics: while Conformer has a computational overhead of $400/500ms$, CAAT requires $1400/1500ms$ more than its ideal LAAL. The CAAT greedy curves (dotted red) show only a little improvement in latency compared to the beam search (solid red), suggesting that its higher computational cost does not depend on the generation strategy but on other factors like its complex and more computationally expensive architecture.

All in all, we can say that, **compared to the state of the art in SimulST, the lower performance of our offline-trained Conformer at low latency regime is balanced by consistently higher BLEU scores at medium and high latency.**

CONCLUSIONS

To reduce the potentially large amount of experiments usually performed to build SimulST models, we explored the use of a single offline-trained model to serve both offline and simultaneous tasks. Through comparison with native SimulST systems, we showed that our offline-trained model can be successfully used in real-time, achieving comparable or even better results. To further enhance its performance, we investigated the adoption of consolidated techniques and emerging architectures from offline research, showing consistent improvements also in the simultaneous scenario. The benefits of offline training indicate the potential of applying this method without the need for any additional training or adaptation. Besides facilitating system deployment, another important advantage of building and reusing one single model to rule both tasks is the drastic reduction of the carbon footprint of ST training (by a factor of 9 in our evaluation setting). This represents an important step in response to rising concerns about the AI energy consumption and environmental impact toward more sustainable development.

As regards SimulST evaluation, the differences between results computed with non-computationally and computationally aware latency metrics suggest that including

computational time in the measurements heavily influences the outcomes of system comparisons. In our particular case, the differences in latency between the offline-trained models and the state of the art observed in terms of the non-computationally aware LAAL metric become smaller when considering its computational aware version. Although lower latency is theoretically reached by the state-of-the-art CAAT model, this comes at the cost of a more complex and computationally expensive architecture that shows its limitations at inference time. We therefore invite the SimulST community to use computationally aware metrics for more sound evaluations, referring to ideal metrics only in the absence of similar testing assets, as machines with comparable computational power.

APPENDIX

3.2.1.9.1 Models Architecture

Transformer The models used in Section 3.2.1.5 are based on 12 encoder and 6 decoder layers of Transformer (Vaswani et al., 2017) architecture. The embedding dimension is set to 256, the number of attention heads to 4 and the feed-forward embedding dimension to 2048, both in the encoder and in the decoder. The number of parameters is $\sim$ 32.4M We use Fairseq (Ott et al., 2019) library for all the trainings. The *wait-k* with fixed word detection strategy was already present in the Fairseq library, while we implemented the adaptive one.

We use the hyper-parameters of (Ma et al., 2020b) for all the trainings of the Transformer-based model. We use a unigram SentencePiece model (Kudo and Richardson, 2018) for the target language vocabulary of size 8,000 Di Gangi et al. (2020). For the source language vocabulary of size 5,000 we use a BPE SubwordNMT model (Sennrich et al., 2016) with Moses tokenizer (Koehn et al., 2007). The reason for which we used SubwordNMT instead of SentencePiece lies in the strategy used for determining the end of a word, which is crucial for simultaneous inference. While SentencePiece uses the character "_" at the beginning of a new word, SubwordNMT appends "@@" to any token that does not represent the end of a word. Thus, SentencePiece units require the generation of the first token of the next word to determine if the current word is over while SubwordNMT units do not. For instance, the sentence "this is a phrase", is encoded into SentencePiece units as "_th is _is _a _ph rase". As such, to determine if "_th is" is a complete word, we have to wait for the next word with the "_" character at the beginning, that is "_is". Instead, with SubwordNMT we have "th@@ is is a

ph@@rase ", and we do not need to receive "is" to determine that "th@@ is" is finished.

We select the best checkpoint based on the loss and early stop the training if the loss does not improve for 10 epochs. We trained the system for 100 epochs at maximum. At the end of the training, we make the average of the 7 checkpoints around the best one.

For the inference part, we use the SimulEval tool (Ma et al., 2020a) as in (Ma et al., 2020b) with the additional `force_finish` tag that forces the model to generate text until the source speech has been completely ingested, i.e. to ignore the end of sentence token if predicted before the end of an utterance. In case of *wait-k* with adaptive word detection, we also force the model to predict the successive most probable token if the end of sentence is predicted (that we called `avoid_eos_while_reading`), while for the fixed we found that it degrades the performance. The detection is taken every average word duration, that is every $280ms$, as estimated by Ma et al. (2020b) in the MuST-C dataset.

Conformer For the Conformer model, we build an architecture similar to Inaguma et al. (2021a), we use 12 Conformer encoder layers and 6 Transformer decoder layers. The number of parameters is $\sim$ 35.7M. We use the same embedding dimension of our Transformer-based architecture, 4 attention encoder heads and 8 attention decoder heads. For the Conformer Feed-Forward layer, Attention layer, and Convolution layer, we use 0.1 as dropout. We use a kernel size of 31 for the point- and depth-wise convolutions of the Convolution layer. The vocabularies are the same as the Transformer-based, as well as the selection of the checkpoint. At inference time, the `force_finish` tag is used with the `avoid_eos_while_reading` for both the word detection strategies.

Machine Translation The MT model used to generate the target for the KD was trained on OPUS datasets (Tiedemann, 2016). It is a plain Transformer with 16 attention heads and 1024 features in encoder/decoder embeddings, resulting in 212M parameters. The English→German MT scores 32.1 BLEU and the English→Spanish MT scores 35.8 BLEU on MuST-C tst-COMMON.

3.2.1.9.2 Under-generation Statistics

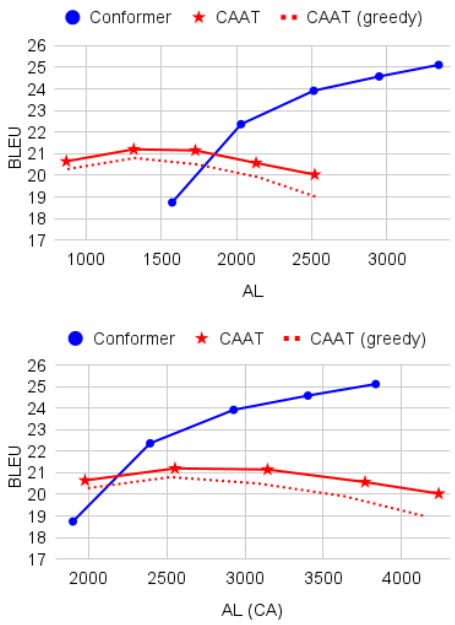
In Section 3.2.1.7, while discussing the en-de curves of Figure 3.7, we highlighted a performance degradation of CAAT at higher latency regimes. In fact, during our experiments, we observed that CAAT tends to generate shorter sentences as the value of k increases. This behaviour becomes apparent in Table 3.3, where we report the word

length difference between the generated hypotheses and the corresponding references. For en-de, CAAT exhibits a strong tendency to under-generate (indicated by negative values) at high latency and this is presumably the reason why we observed the BLEU drop.

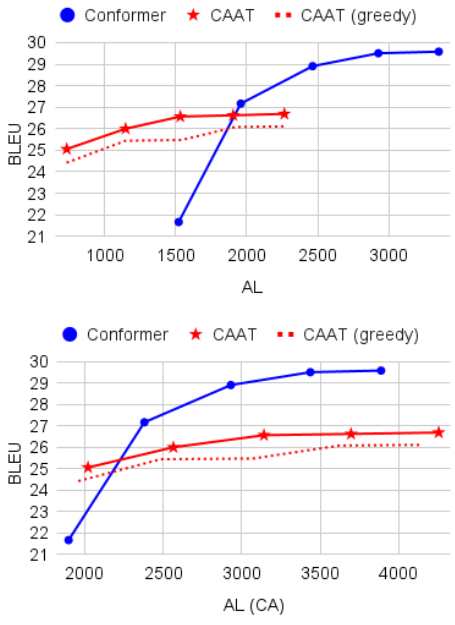
English→German					
Model	k=3	k=5	k=7	k=9	k=11
Conformer	-1	-0.94	-0.93	-0.77	-0.63
CAAT	0.47	-0.3	-0.79	-1.26	-1.55
English→Spanish					
Model	k=3	k=5	k=7	k=9	k=11
Conformer	0.48	0.49	0.53	0.74	0.80
CAAT	1.57	0.96	0.61	0.35	0.18

Table 3.3: Average word length difference w.r.t. the reference. Positive values indicate exceeding words, negative values indicate missing words.

3.2.1.9.3 Average Lagging



(a) English → German



(b) English → Spanish

Figure 3.8: AL/AL_{CA}-BLEU curves of our offline-trained Conformer and CAAT models.

3.2.2 PAPER #2

Attention as a Guide for Simultaneous Speech Translation

INTRODUCTION

In simultaneous speech translation (SimulST), systems have to generate translations incrementally while concurrently receiving audio input. This requirement poses a significant challenge since the need of generating high-quality outputs has to be balanced with the need to minimize their latency, i.e. the time elapsed (lagging) between when a word is uttered and when it is actually translated by the system.

In direct SimulST systems (Bérard et al., 2016; Weiss et al., 2017),[11] the balance between output quality and latency is managed by a *decision policy*, which is the strategy for determining, at each time step, whether to emit a partial translation or to wait for additional audio input. Decision policies can be divided into two categories: *fixed* and *adaptive*. Fixed policies are usually based on simple heuristics (Ma et al., 2019a), while adaptive policies take into account the actual input content to make the decisions (Zheng et al., 2020). Recent works (Liu et al., 2021b; Zaidi et al., 2021, 2022; Zhang and Feng, 2022) proved the superiority of adaptive policies over fixed ones. However, a major limitation of these policies is that they require training *ad-hoc* and complex SimulST architectures, which results in high computational costs.

Computational costs are also inflated by the common practice of simulating the simultaneous test conditions by providing partial input during training to avoid the quality drops caused by the mismatch between training and test conditions (Ren et al., 2020; Ma et al., 2020b, 2021b; Han et al., 2020; Zeng et al., 2021; Liu et al., 2021a; Zaidi et al., 2021, 2022). This practice is independent of the decision policy adopted, and typically requires dedicated trainings for each latency regime. To mitigate this issue, offline-trained ST systems have been employed for simultaneous inference (Liu et al., 2020b; Chen et al., 2021; Nguyen et al., 2021) and, along this direction, Papi et al. (2022b) demonstrated that dedicated trainings simulating the inference conditions are not necessary since offline-trained systems outperform those specifically trained for SimulST. The effectiveness of using offline-trained ST models for simultaneous inference

[11] In this paper, we focus on direct models that exhibit lower latency and better performance compared to traditional cascade architectures composed of separate automatic speech recognition and machine translation components (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022).

has been also confirmed by the last IWSLT 2022 evaluation campaign (Anastasopoulos et al., 2022), where the winning submission to the SimulST task (Polák et al., 2022) is an offline model exploiting the Local Agreement policy by Liu et al. (2020b). However, despite its good results, this policy relies on a strategy (the generation of two consecutive hypotheses prior to starting the emission) that has a significant impact on latency. This raises the need for effective policies that *i)* are adaptive, *ii)* are directly applicable to offline ST models, and *iii)* achieve low latency at low computational costs.

Towards these objectives, we propose EDAtt (**E**ncoder-**D**ecoder **Att**ention),[12] a novel adaptive policy for SimulST that leverages the encoder-decoder attention patterns of an offline-trained ST model to decide when to emit partial translations. In a nutshell, our idea is that the next word of the partial hypothesis at a given time step is safely emitted only if the system does not attend to the most recent audio frames, meaning that the information received up to that time step is sufficient to generate that word. Building on this idea, our contributions are summarized as follows:

- We introduce EDAtt, a novel adaptive decision policy for SimulST, which guides offline-trained ST models during simultaneous inference by looking at the attention patterns dynamically computed from the audio input over time;
- We show that EDAtt outperforms the Local Agreement policy applied to the same offline ST models at almost all latency regimes, with computational-aware average lagging (AL_CA) reductions up to 1.4s for German and 0.7s for Spanish on MuST-C (Cattoni et al., 2021);
- We show that EDAtt also outperforms the state-of-the-art CAAT architecture (Liu et al., 2021b), especially in terms of AL_CA, with gains of up to 7.0 BLEU for German and 4.0 BLEU for Spanish.

BACKGROUND

In terms of architectural choices, Transformer (Vaswani et al., 2017) and its derivatives (Gulati et al., 2020; Chang et al., 2020; Papi et al., 2021b; Burchi and Vielzeuf, 2021; Kim et al., 2022; Andrusenko et al., 2022) are the *de-facto* standard both in offline and simultaneous ST (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022).

A generic Transformer model is composed of an encoder, whose role is to map the input speech sequence $\mathbf{X} = [x_1, ..., x_n]$ into an internal representation, and a decoder,

[12] Code, outputs and offline ST models used for our experiments are released under Apache License 2.0 at: `https://github.com/hlt-mt/fbk-fairseq`.

whose role is to generate the output textual sequence $\mathbf{Y} = [y_1, ..., y_m]$ by exploiting the internal representation in an auto-regressive manner (Graves, 2013), that is by consuming the previously generated output as additional input when generating the next one.

The encoder and the decoder are composed of a stack of identical blocks, whose components may vary depending on the particular Transformer-based architecture, although they all share the same dot-product attention mechanism (Chan et al., 2016). In general, the attention is a function that maps a query matrix Q and a pair of key-value matrices (K, V) to an output matrix (Bahdanau et al., 2016). The output is obtained as a weighted sum of V, whose weights are computed through a compatibility function between Q and K that, in the case of the scaled dot-product attention used in the original Transformer formulation, is:

$$A(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimension of K. The attention A is computed on h heads in parallel, each applying learned linear projections W^Q, W^K, and W^V to the Q, K, and V matrices. These representations are then concatenated and projected using another learned matrix W^O, resulting in the final output:

$$\begin{aligned}&\text{Multihead}(Q, K, V) = \\ &\quad \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W_O\end{aligned}$$

where $\text{head}_i = A(QW_i^Q, KW_i^K, VW_i^V)$.

In the encoder layers, Q, K, and V are computed from the same speech input sequence $\mathbf{X}$, realizing the so-called *self*-attention $A_{\text{self}}(\mathbf{X})$. Differently, in the decoder layer, two types of attention are computed sequentially: self-attention, and *encoder-decoder* (or cross) attention. In the encoder-decoder attention, Q comes from the previous decoder layer (or directly from the previously generated output $\mathbf{Y}$, in the case of the first decoder layer) while K and V come from the output of the encoder, hence the matrix can be expressed as $A_{\text{cross}}(\mathbf{X}, \mathbf{Y})$. In this work, we only exploit the encoder-decoder attention matrix to guide the model during simultaneous inference. Therefore, we use the notation A instead of A_{cross} for simplicity, and henceforth refer to this matrix as the encoder-decoder representation of a specific decoder layer d considering the attention head h.

EDATT POLICY

We propose to exploit the information contained in the encoder-decoder attention matrix of an offline ST model during inference to determine whether to wait for additional audio input or emit a partial translation. The use of attention as the core mechanism of our policy is motivated by related works in machine translation (MT) and language modeling, which prove that attention scores can encode syntactic dependencies (Raganato and Tiedemann, 2018; Htut et al., 2019) and language representations (Lamarre et al., 2022), as well as align source and target tokens (Tang et al., 2018; Zenkel et al., 2019; Garg et al., 2019; Chen et al., 2020). We posit (and demonstrate in Section 3.2.2.5) that this encoder-decoder attention relationship between source audio and target tokens also exists in offline ST models, and can be used to guide them during simultaneous inference.

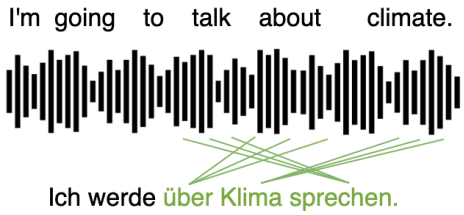
Our approach builds on the following hypothesis (see Figure 3.9): at each time step, if the attention is focused towards the end of the input audio sequence (1), the system will probably need more information to correctly produce the current output candidate. On the contrary (2), if the attention concentrates on early audio frames (far enough from the last received ones), the current output candidate can be safely emitted because the early encoded information is sufficient. Accordingly, the model will continue to emit the next token of the partial hypothesis until the above condition is verified, that is until its encoder-decoder attention scores do not focus towards the end of the received speech segment. The rationale is that if the encoder-decoder attention of the predicted token points to the most recent speech information – i.e. attention scores are higher towards the last audio frames received – this information could be incomplete and therefore still insufficient to generate that token.

More formally, at each time step t, EDAtt determines whether to emit the next token y_j, given the previously generated tokens $\mathbf{Y}_{j-1} = [y_1, ..., y_{j-1}]$ and the partial audio input sequence $\mathbf{X}_t$, by looking at the sum of the last λ encoder-decoder attention weights of the vector $A_j(\mathbf{X}_t, \mathbf{Y}_{j-1})$. Specifically, y_j is emitted if:

$$\sum_{i=t-\lambda}^{t} A_{i,j}(\mathbf{X}_t, \mathbf{Y}_{j-1}) < \alpha, \quad \alpha \in (0, 1) \tag{3.11}$$

where α is a hyperparameter that controls the quality-latency trade-off: lower values of α increase the latency, as they reduce the possibility to satisfy Equation 3.11 (i.e. the sum of the last λ encoder-decoder attention weights will likely exceed α), and vice versa. When Equation 3.11 is satisfied, y_j is emitted and the same process is repeated for y_{j+1},

(1) When the first speech segment is received, the partial hypothesis "*Ich werde*" is emitted since the attention is not concentrated towards the end of the segment while "*reden.*" is not since the attention is all concentrated on the last frames.



(2) When the second speech segment is received, the new partial hypothesis "*über Klima sprechen.*" is emitted since the attention is not concentrated towards the end of the segment.

Figure 3.9: Example of the EDAtt policy. Links indicate where the attention weights point to.

and so on. The process continues until we reach the token y_{j+w} for which Equation 3.11 is no longer verified. At that point, the emission is stopped and the total number of tokens emitted at time step t is w.

EXPERIMENTAL SETTINGS

3.2.2.4.1 Data

To be comparable with previous works (Ren et al., 2020; Ma et al., 2020b; Zeng et al., 2021; Liu et al., 2021b; Papi et al., 2022b; Zhang and Feng, 2022), we train our models on MuST-C en→{de, es} (Cattoni et al., 2021). The choice of the two target languages is also motivated by their different word ordering: Subject-Object-Verb (SOV) for German and Subject-Verb-Object (SVO) for Spanish. This opens the possibility of validating our approach on target-language word orderings that are respectively different and similar with respect to the English (i.e. SVO) source audio. We also perform data augmentation by applying sequence-level knowledge distillation (Kim and Rush, 2016; Gaido et al., 2021b, 2022a) as in (Liu et al., 2021b; Papi et al., 2022b), for which the transcripts of MuST-C en→{de, es} are translated with an MT model (more details can be found in Appendix 3.2.2.9.1) and used together with the gold reference during training. Data statistics are given in Appendix 3.2.2.9.2.

3.2.2.4.2 Architecture and Training Setup

For our experiments, we use the bug-free implementation by Papi et al. (2023c) of the Conformer-based encoder-decoder model for ST (Guo et al., 2021). The offline model is made of 12 Conformer encoder layers (Gulati et al., 2020) and 6 Transformer decoder layers ($d_{max} = 6$) with a total of ∼115M parameters. Each encoder/decoder layer has 8 attention heads ($h_{max} = 8$). The input is represented as 80 audio features extracted every $10ms$ with sample window of 25 and processed by two 1D convolutional layers with stride 2 to reduce its length by a factor of 4 (Wang et al., 2020a). Utterance-level Cepstral Mean and Variance Normalization (CMVN) and SpecAugment (Park et al., 2019) are applied during training. Detailed settings are described in Appendix 3.2.2.9.1.

3.2.2.4.3 Inference and Evaluation

We use the SimulEval tool (Ma et al., 2020a) to simulate simultaneous conditions and evaluate all the models. For our policy, we vary α of Equation 3.11 in the range $[0.6, 0.4, 0.2, 0.1, 0.05, 0.03]$ and set the size of the speech segment to $800ms$. During inference, the features are computed on the fly and CMVN normalization is based on the global mean and variance estimated on the MuST-C training set. All inferences are performed on a single NVIDIA K80 GPU with 12GB memory as in the IWSLT Simultaneous evaluation campaigns.

We use sacreBLEU (Post, 2018)[13] to evaluate translation quality, and Average Lagging (Ma et al., 2019a) – or AL – to evaluate latency, as in the default SimulEval evaluation setup. As suggested by Ma et al. (2020b), for our comparisons with other approaches we also report computational-aware average lagging (AL_CA), which measures the real elapsed time instead of the ideal one considered by AL, thus giving a more realistic latency measure when the system operates in real time. Its computation is also provided by SimulEval.

3.2.2.4.4 Terms of Comparison

We conduct experimental comparisons with the state-of-the-art architecture for SimulST (CAAT) and, respectively, the current best (Local Agreement) and the most widely used (Wait-k) policies that can be directly applied to our offline ST systems for simultaneous inference. In detail:

[13] BLEU+case.mixed+smooth.exp+tok.13a+version.1.5.1

Cross Attention Augmented Transformer (CAAT) – the state-of-the-art architecture for SimulST (Liu et al., 2021b), winner of the IWSLT 2021 SimulST task (Anastasopoulos et al., 2021). Inspired by the Recurrent Neural Network Transducer (Graves, 2012), it is made of three Transformer stacks: the encoder, the predictor, and the joiner. These three elements are jointly trained to optimize translation quality while keeping latency under control. We train and evaluate the CAAT model using the code provided by the authors,[14] and on the same data used for our offline ST model.

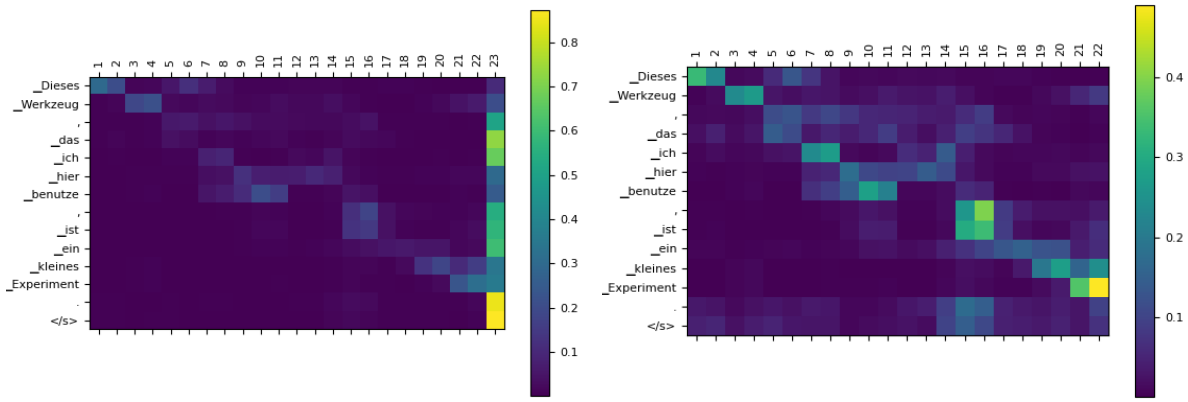
Local Agreement (LA) – the state-of-the-art decision policy introduced by Liu et al. (2020b), and used by the winning system at IWSLT 2022 (Anastasopoulos et al., 2022). It consists of generating a partial hypothesis from scratch each time a new speech segment is added, and emitting it – or part of it – if it coincides with one of those generated in the previous l time steps, where l is a hyperparameter. Since Liu et al. (2020b) empirically found that considering only the most recent previously generated tokens ($l = 1$) as memory works better, we adopt the same strategy to apply this policy.

Wait-k – the simplest and most widely used decision policy in SimulST (Ren et al., 2020; Ma et al., 2020b; Zeng et al., 2021). It consists in waiting for a fixed number of words (k) before starting to emit the translation, and then proceeding by alternating waiting and writing operations. Since in SimulST the information about the number of words is not explicitly contained in the audio input, a word detection strategy is used to determine this information. Detection strategies can be fixed when it is assumed that each word has a pre-defined fixed duration, or adaptive when the information about the number of words is inferred from the audio content. Following Papi et al. (2022b), we adopt a CTC-based adaptive word detection strategy to detect the number of words. In addition, to be comparable with the other approaches, we employ beam search to generate each token.

ATTENTION ANALYSIS

To validate our hypothesis and study the feasibility of our method, we start by exploring the encoder-decoder attention matrices of the offline trained models. We proceed as follows: first, by visualizing the attention weights, we check for the existence of patterns that could be exploited during simultaneous inference. Then, we analyze the performance of the EDAtt policy to discover the best value of λ, the decoder layer d,

[14] `https://github.com/danliu2/caat`



(a) Unfiltered (b) Filtered

Figure 3.10: Encoder-decoder attention scores on a random sample of the MuST-C en→de dev set, before (a) and after (b) the filtering of the last frame from the attention matrix.

and the attention head h from which to extract the attention scores that better balance the quality-latency trade-off.

Do attention patterns exist also in ST? To answer this question, we conducted an analysis of the encoder-decoder matrices obtained from the MuST-C en-de dev set. Through the visualization of attention weights, we observed a consistent phenomenon across our two language directions (en→{de, es}): the attention weights concentrate on the last frame, regardless of the input length, as shown in Figure 2a. This behaviour has already been observed in prior works on attention analysis, showing that attention often concentrates on the initial or final token (Clark et al., 2019; Kovaleva et al., 2019; Kobayashi et al., 2020; Ferrando et al., 2022), with up to 97% of attention weights being allocated to these positions. As this might hinder the possibility to effectively visualize attention patterns, similarly to (Vig and Belinkov, 2019), we filtered out the last frame from the attention matrix and then re-normalized it. In this way, as shown in Figure 2b, we obtained a clear pseudo-diagonal pattern compared to the previous unfiltered representation. Such correspondence emerging from the encoder-decoder attention scores after the removal of the last frame indicates a relationship between the source audio frames and target translation texts that can be exploited by our adaptive attention-based policy during simultaneous inference.

What is the optimal value of λ? To find the best number of frames (λ) on which to apply Equation 3.11, we analyse the behavior of EDAtt by varying α and

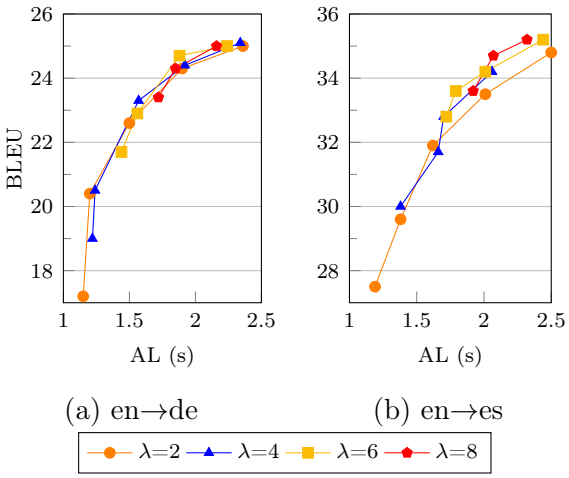


Figure 3.11: Effect of λ on MuST-C en→{de, es} dev set. We visualize the results with AL $\leq 2.5s$.

setting $\lambda \in [2, 4, 6, 8]$.[15] For this analysis, we extract the attention scores from the 5^{th} decoder layer ($d = 5$) by averaging across the matrices obtained from each attention head ($h = [1, ..., 8]$) in accordance with the findings of (Garg et al., 2019) about the layer that best represents word alignment. We perform the analysis on the MuST-C dev set for both language pairs, and present the results in Figure 3.11. As we can see, as the value of λ increases, the curves shift towards the right, indicating an increase in latency. This means that, consistently across languages, considering too many frames towards the end ($\lambda \geq 6$) affects latency with little effect on quality. Since $\lambda = 2$ yields the lowest latency (AL $\approx 1.2s$) in both languages, and especially in Spanish, we select this value for the following experiments. This outcome is noteworthy as it demonstrates that, at least in our settings, the same optimal value of λ applies to diverse target languages with different word ordering. However, this might not hold for different source and/or target languages, advocating for future explorations as discussed in the Limitations section.

What is the best layer? After determining the optimal value of λ, we proceed to analyze the EDAtt performance by varying the decoder layer from which the encoder-decoder attention is extracted. We conduct this study by using $\lambda = 2$, as previously determined to be the optimal value for both languages. In Figure 3.12, we present the SimulST results (in terms of AL-BLEU curves) for each decoder layer $d = [1, ..., 6]$.[16]

[15] We do not report the experiments with $\lambda = 1$ since we found that it consistently degrades translation quality. We also experimented with different ways to determine λ, such as using a percentage instead of a fixed number, but none of them yielded significant differences.

[16] We also tried to make the average of the encoder-decoder attention matrices of each layer but this led to worse results.

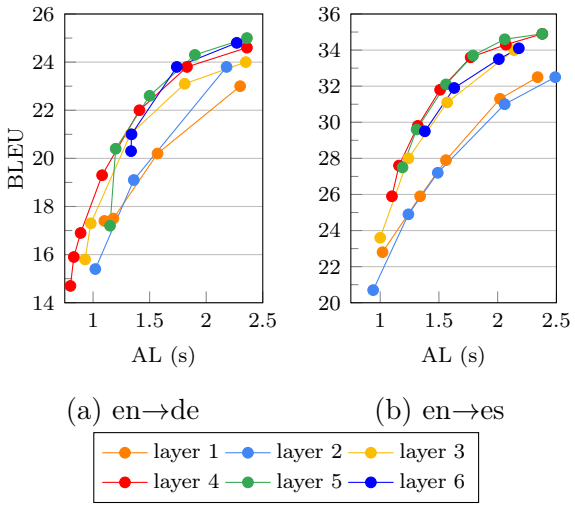


Figure 3.12: SimulST results on MuST-C dev set en→{de, es} for each decoder layer d. We visualize the results with AL $\leq 2.5s$.

As we can see, in both languages, Layers 1 and 2 consistently perform worse than the other layers. Also, Layer 3 achieves inferior quality compared to Layers ≥ 4, especially at medium-high latency (AL $\geq 1.2s$) despite performing better than Layers 1 and 2. This aligns with the findings of Garg et al. (2019), which observed inferior performance by the first three layers in the alignment task for MT models. Concerning Layer 6, both graphs show that the curves cannot achieve lower latency, starting at around $1.5s$ of AL. This phenomenon is also valid for Layer 5 compared to Layer 4, although being much less pronounced. We also observe that Layer 5 achieves the best performance at higher latency on both languages. However, since Layers 5 and 6 never achieve low latency (AL never approaches $1.2s$), we can conclude that the optimal choice for the simultaneous scenario is Layer 4. This is in line with Lamarre et al. (2022), which indicates the middle layers as the best choice to provide accurate predictions for language representations. As a consequence, we will use $d = 4$ for the subsequent experiments with EDAtt.

Would a single attention head encode more useful information?

According to prior research examining the usefulness of selecting a single or a set of attention heads to perform natural language processing and translation tasks (Jo and Myaeng, 2020; Behnke and Heafield, 2020; Gong et al., 2021), we also investigate the behavior of the EDAtt policy by varying the attention head h from which the encoder-

decoder attention matrix A is extracted. In Table 3.4,[17] we present the results obtained from each attention head $h = [1, ..., 8]$.[18] Firstly, we observe that many heads are unable to achieve low latency, particularly for Spanish. Furthermore, there is no consensus on the optimal head among languages or at different latencies (e.g. Head 6 is the best in Spanish at 1.6s, but it does not achieve lower latency). However, we notice that the average across all heads (last row) has an overall better performance compared to the encoder-decoder matrices extracted from each individual head, and this holds true for both languages. Consequently, we choose to compute the average over the attention heads to apply our EDAtt policy in order to achieve a better quality-latency trade-off for SimulST.

Head	**en→de**			**en→es**		
	1.2s	1.6s	2s	1.2s	1.6s	2s
Head 1	17.6	19.2	20.5	27.6	30.8	32.1
Head 2	19.0	21.9	23.4	-	31.9	33.9
Head 3	-	22.3	23.9	27.2	29.8	31.1
Head 4	-	21.5	23.3	-	28.4	30.7
Head 5	19.2	22.2	23.8	-	30.9	32.5
Head 6	18.7	21.2	22.7	-	32.0	33.3
Head 7	-	21.9	23.5	-	30.8	32.6
Head 8	19.2	20.7	21.6	-	31.7	33.9
Average	**20.3**	**22.8**	**24.0**	**28.6**	**32.4**	**34.1**

Table 3.4: BLEU scores on MuST-C dev set en→{de, es} for each attention head h of Layer 4. Latency (AL) is reported in seconds. "-" means that the BLEU value is not available or calculable. The last row represents the numerical values of Layer 4 curves of Figure 3.12 obtained by averaging across all 8 heads.

RESULTS

3.2.2.6.1 Comparison with Other Approaches

For the comparison of EDAtt with the SimulST systems described in Section 3.2.2.4.4, we report in Figure 3.13 both AL (solid curves) and AL_CA (dashed curves) as latency measures to give a more realistic evaluation of the performance of the systems in real

[17] A tabular format is used instead of AL-BLEU curves as many parts of the curves are indistinguishable from each other. AL = 1.2s is the first latency measure reported because it is the minimum value spanned by the head-wise curves, and AL = 2s is the last one since increasing latency above this value does not significantly improve translation quality (BLEU).

[18] Since obtaining a specific latency in seconds is not possible with this method, we interpolate the previous and successive points to estimate the BLEU value, when needed.

time, as recommended in (Ma et al., 2020b; Papi et al., 2022b). Results with other metrics, DAL (Cherry and Foster, 2019) and LAAL (Papi et al., 2022a), are provided in Appendix 3.2.2.9.3 for completeness. Numeric values for all the plots are presented in Section 3.2.2.9.4. For our policy, we extract the encoder-decoder attention matrix from Layer 4 ($d = 4$), average the weights across heads, and set $\lambda = 2$ as it was found to be the optimal setting on the MuST-C dev set for both languages, as previously discussed in Section 3.2.2.5.

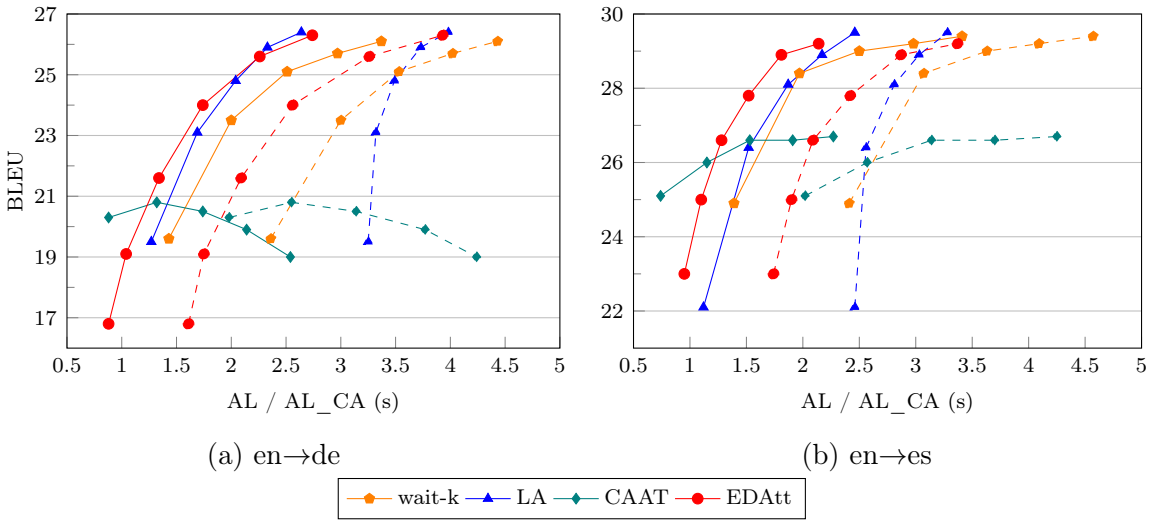
Quality-latency curves for en→de and en→es show similar trends. The EDAtt policy achieves better overall results compared to the LA and wait-k policies applied to offline ST models. EDAtt consistently outperforms the wait-k policy, with gains ranging from 1.0 to 2.5 BLEU for German and 1.0 to 3 for Spanish, when considering both ideal (AL) and computationally aware (AL_CA) latency measures. Additionally, it is able to achieve lower latency, as the starting point of the wait-k policy is always around $1.5s$, while EDAtt starts at $1.0s$. In comparison to the LA policy, we observe an AL_CA reduction of up to $1.4s$ for German and $0.7s$ for Spanish. Moreover, the computational overhead of EDAtt is consistently lower, $0.9s$ on average between languages, against $1.3s$ of LA. Therefore, the computational cost of our policy is 30% lower compared to the LA policy. Additionally, EDAtt outperforms LA at almost every latency, with gains up to 2.0 BLEU for German and 3.0 for Spanish.

Compared with CAAT, when ideal latency is considered (solid curves), we notice that EDAtt achieves higher quality at medium-high latency (AL $\geq 1.2s$), with BLEU gains up to 5.0 points for German and 2.0 for Spanish. When AL $< 1.2s$, instead, there is a decrease in performance with BLEU drops ranging from 1.5 to 4.0 for German and 1.0 to 2.5 for Spanish. However, when considering the realistic computational-aware latency measure AL_CA (dashed curves), we observe that the EDAtt curves are always to the left of those of the CAAT system, indicating that our policy always outperforms it with BLEU gains up to 6.0 points for German and 2.0 for Spanish.

In light of this, we can conclude that EDAtt achieves new state-of-the-art results in terms of computational-aware metrics, while also being superior at medium-high latency when considering the less realistic computational-unaware measure.

3.2.2.6.2 Effects of Accelerated Hardware

To further investigate the computational efficiency of EDAtt, we conducted experiments on all the systems described in Section 3.2.2.4.4 using a highly accelerated GPU, an NVIDIA A40 with 48GB memory, during simultaneous inference.



(a) en→de (b) en→es

Figure 3.13: Comparison with the SimulST systems described in Section 3.2.2.4.4 on MuST-C en→{de, es} tst-COMMON. Solid curves represent AL, dashed curves represent AL_CA.

Figure 3.14 reports the results in terms of quality-latency trade-off. When comparing the curves with the computationally aware ones in Figure 3.13 (dashed), it can be observed that the LA policy seems to benefit more from the use of expensive accelerated hardware, with a latency reduction of 0.5-1s. However, this reduction is not sufficient to reach a latency lower than 2s with this policy. Considering the other systems, both wait-k and CAAT curves show a slight left shift (by less than 0.5s), similar to EDAtt.[19]

In conclusion, our policy proved to be superior even when using accelerated and expensive hardware, further strengthening the previously discussed findings. Moreover, these results indicate that there are no significant differences between the systems when using less or more accelerated GPU hardware and advocate for the wider use of computationally aware metrics in future research.

RELATED WORKS

The first policy for SimulST was proposed by Ren et al. (2020) and is derived from the wait-k policy (Ma et al., 2019a) developed for simultaneous *text-to-text* translation. Most of subsequent studies have also adopted the wait-k policy (Ma et al., 2020b; Han et al., 2020; Chen et al., 2021; Zeng et al., 2021; Karakanta et al., 2021b; Nguyen

[19]Despite the benefits in terms of quality-latency trade-off, the significantly higher costs of the A40 GPU over the K80 GPU (4.1 vs 0.9 USD/h in Amazon Web Services, `https://aws.amazon.com/it/ec2/pricing/on-demand/`) makes unlikely that such a GPU will soon be of widespread use for simultaneous inference.

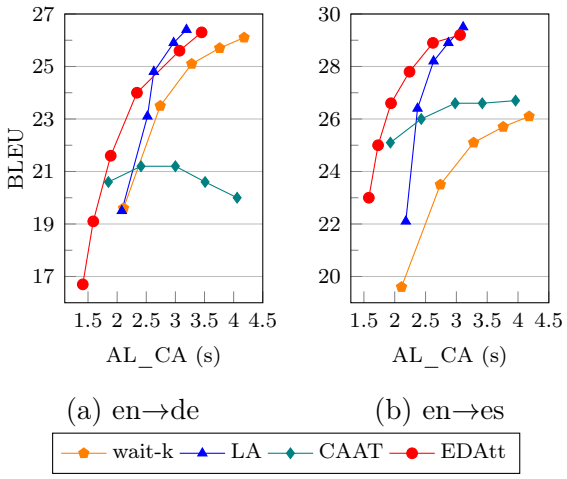


Figure 3.14: Effect of using NVIDIA A40 GPU on MuST-C en→{de, es} tst-COMMON considering all the systems of Section 3.2.2.4.4. Results are computationally aware.

et al., 2021; Papi et al., 2022b). In parallel, several strategies have been developed to directly learn the best policy during training by means of *ad-hoc* architectures (Ma et al., 2021b; Liu et al., 2021a,b; Chang and Lee, 2022) and training procedures aimed at reducing latency (Liu et al., 2021a,b; Zaidi et al., 2021, 2022; Chang and Lee, 2022; Zhang and Feng, 2022; Omachi et al., 2023). The latter adaptive policies obtained better performance according to the most recent results observed in (Anastasopoulos et al., 2021, 2022). We define our policy as adaptive as well, as it relies on the encoder-decoder attention mechanism, whose dynamics are influenced by the audio input that increases incrementally over time. However, EDAtt completely differs from prior works on adaptive policies that exploit attention (Zaidi et al., 2021, 2022; Chang and Lee, 2022; Zhang and Feng, 2022) because is the first policy that does not require influencing the behaviour of the attention weights through dedicated training strategies, therefore being directly applicable to offline-trained ST models. By doing so, we realize *i)* an adaptive policy, *ii)* directly applicable to offline-trained ST models, *iii)* which achieves low latency at low computational costs.

CONCLUSIONS

After investigating the encoder-decoder attention behavior of offline ST models, we presented EDAtt, a novel adaptive decision policy for SimulST that guides an offline ST model to wait or to emit a partial hypothesis by looking at its encoder-decoder attention weights. Comparisons with state-of-the-art SimulST architectures and decision policies reveal that, at lower computational costs, EDAtt outperforms the others at

almost every latency, with translation quality gains of up to 7.0 BLEU for en→de and 4.0 BLEU for en→es. Moreover, it is also capable of achieving a computational-aware latency of less than $2s$ with a reduction of 0.7-1.4s compared to existing decision policies applied to the same offline ST systems.

APPENDIX

3.2.2.9.1 Training Settings

We use 512 as embedding size and 2,048 hidden neurons in the feed-forward layers both in the encoder and in the decoder. We set dropout at 0.1 for feed-forward, attention, and convolution layers. Also, in the convolution layer, we set 31 as kernel size for the point- and depth-wise convolutions. The vocabularies are based on SentencePiece (Sennrich et al., 2016) with dimension of 8,000 (Di Gangi et al., 2020) for the target side (de, es) and of 5,000 (Wang et al., 2020a) for the source side (en). We optimize with Adam (Kingma and Ba, 2015) by using the label-smoothed cross-entropy loss with 0.1 as smoothing factor (Szegedy et al., 2016). We employ Connectionist Temporal Classification – or CTC – (Graves et al., 2006) as auxiliary loss to avoid pre-training (Gaido et al., 2022e) and also to compress the input audio, reducing RAM consumption and speeding up inference (Gaido et al., 2021a). The learning rate is set to $5 \cdot 10^{-3}$ with Noam scheduler (Vaswani et al., 2017) and warm-up steps of 25k. We stop the training after 15 epochs without loss decrease on the dev set and average 7 checkpoints around the best (best, three preceding, and three succeeding). Trainings are performed on 4 NVIDIA A40 GPUs with 40GB RAM. We set 40k as the maximum number of tokens per mini-batch, 2 as update frequency, and 100,000 as maximum updates (∼23 hours).

The MT models used for knowledge distillation are trained on OPUS (Tiedemann, 2016) en→{de, es} sections and are plain Transformer architectures with 16 attention heads and 1024 embedding features in the encoder/decoder, resulting in ∼212M parameters. We achieve 32.1 and 35.8 BLEU on, respectively, MuST-C tst-COMMON German and Spanish.

3.2.2.9.2 Data Statistics

MuST-C training data (train set) has been filtered: samples containing audio longer than 30s are discarded to reduce GPU computational requests. The total number of samples used during our trainings is shown in Table 3.5.

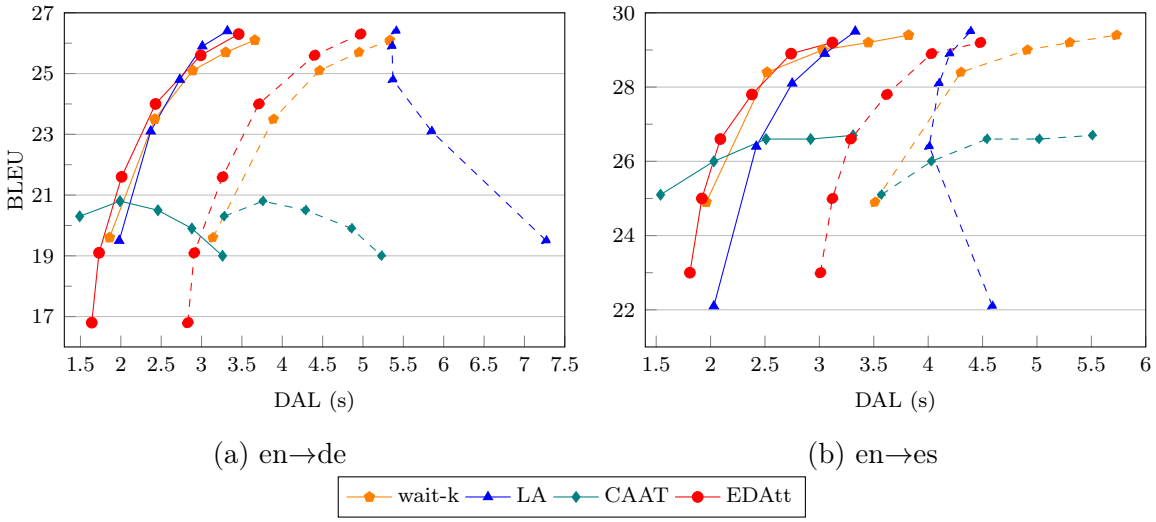
split	en→de	en→es
train	225,277*	260,049*
dev	1,423	1,316
tst-COMMON	1,422	1,315

Table 3.5: Number of samples for each split of MuST-C. * means this number doubled due to the use of KD.

3.2.2.9.3 Main Results with Different Latency Metrics

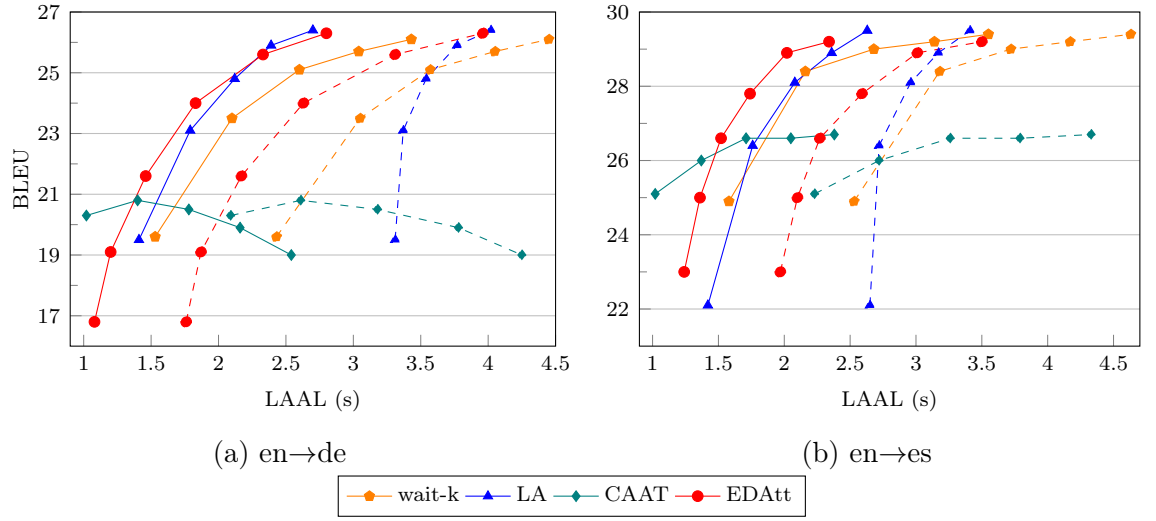
Apart from AL, two metrics can be adopted to measure latency in simultaneous. The first one is the Differentiable Average Lagging – or DAL – (Cherry and Foster, 2019), a differentiable version of AL, and the Length-Adaptive Average Lagging – or LAAL – (Papi et al., 2022a), which is a modified version of AL that accounts also for the case in which the prediction is longer compared to the reference. Figure 3.15 and 3.16 show the results of the systems of Figure 3.13 by using, respectively, DAL and LAAL considering both computational aware (CA) and unaware metrics for German and Spanish. Numeric values are presented in Section 3.2.2.9.4.

As we can see, the results of Figure 3.15 and 3.16 confirm the phenomena found in Section 3.13, indicating EDAtt as the best system among languages and latency values. We observe also that DAL reports higher latency for all systems (it spans from 3 to 7.5s for German and to 5.5s for Spanish), with a counter-intuitive curve for the LA method considering its computational aware version. However, we acknowledge that DAL is less suited than AL/LAAL to evaluate current SimulST systems: in its computation, DAL gives a minimum delay for each emitted word while all the systems considered in our analysis can emit more than one word at once, consequently being improperly penalized in the evaluation.



(a) en→de (b) en→es

Figure 3.15: DAL results for the SimulST systems of Section 3.2.2.4.4. Solid curves represent DAL, dashed curves represent DAL_CA.



(a) en→de (b) en→es

Figure 3.16: LAAL results for the SimulST systems of Section 3.2.2.4.4. Solid curves represent LAAL, dashed curves represent LAAL_CA.

3.2.2.9.4 Numeric Values for Main Results

Table 3.6 on the next page.

en-de							
Policy	**BLEU**	**AL**	**AL_CA**	**LAAL**	**LAAL_CA**	**DAL**	**DAL_CA**
wait-k	19.6	1.43	2.36	1.53	2.43	1.86	3.14
	23.5	2.00	3.00	2.10	3.05	2.42	3.89
	25.1	2.51	3.53	2.60	3.57	2.89	4.46
	25.7	2.97	4.02	3.04	4.05	3.30	4.95
	26.1	3.37	4.43	3.43	4.45	3.66	5.33
LA	19.5	1.27	3.25	1.41	3.31	1.98	7.27
	23.1	1.69	3.32	1.79	3.37	2.37	5.85
	24.8	2.04	3.49	2.12	3.54	2.73	5.37
	25.9	2.33	3.73	2.39	3.77	3.01	5.36
	26.4	2.64	3.98	2.70	4.02	3.32	5.41
CAAT	20.3	0.88	1.98	1.02	2.09	1.49	3.28
	20.8	1.32	2.55	1.40	2.61	1.99	3.76
	20.5	1.74	3.14	1.78	3.18	2.46	4.29
	19.9	2.14	3.77	2.16	3.78	2.88	4.86
	19.0	2.54	4.24	2.54	4.25	3.26	5.23
EDAtt	16.8	0.88	1.61	1.08	1.76	1.64	2.83
	19.1	1.04	1.75	1.20	1.87	1.73	2.91
	21.6	1.34	2.09	1.46	2.17	2.01	3.26
	24.0	1.74	2.56	1.83	2.63	2.43	3.71
	25.6	2.26	3.26	2.33	3.31	2.99	4.40
	26.3	2.74	3.93	2.80	3.96	3.46	4.97
en-es							
Policy	**BLEU**	**AL**	**AL_CA**	**LAAL**	**LAAL_CA**	**DAL**	**DAL_CA**
wait-k	24.9	1.39	2.41	1.58	2.53	1.96	3.51
	28.4	1.97	3.07	2.16	3.18	2.52	4.30
	29.0	2.50	3.63	2.68	3.72	3.03	4.91
	29.2	2.98	4.09	3.14	4.17	3.45	5.30
	29.4	3.41	4.57	3.55	4.63	3.82	5.73
LA	22.1	1.12	2.46	1.42	2.65	2.03	4.59
	26.4	1.52	2.56	1.76	2.72	2.42	4.01
	28.1	1.87	2.81	2.08	2.96	2.75	4.10
	28.9	2.17	3.03	2.36	3.17	3.05	4.20
	29.5	2.46	3.28	2.63	3.41	3.33	4.39
CAAT	25.1	0.74	2.02	1.02	2.23	1.54	3.57
	26.0	1.15	2.57	1.37	2.72	2.03	4.03
	26.6	1.53	3.14	1.71	3.26	2.51	4.54
	26.6	1.91	3.70	2.05	3.79	2.92	5.02
	26.7	2.27	4.25	2.38	4.33	3.31	5.51
EDAtt	23.0	0.95	1.74	1.24	1.97	1.81	3.01
	25.0	1.10	1.90	1.36	2.10	1.92	3.12
	26.6	1.28	2.09	1.52	2.27	2.09	3.29
	27.8	1.52	2.42	1.74	2.59	2.38	3.62
	28.9	1.81	2.87	2.02	3.01	2.74	4.03
	29.2	2.14	3.37	2.34	3.50	3.12	4.48

Table 3.6: Numeric values for the plots presented in Sections 3.2.2.6 and 3.2.2.9.3.

3.2.3 PAPER #3

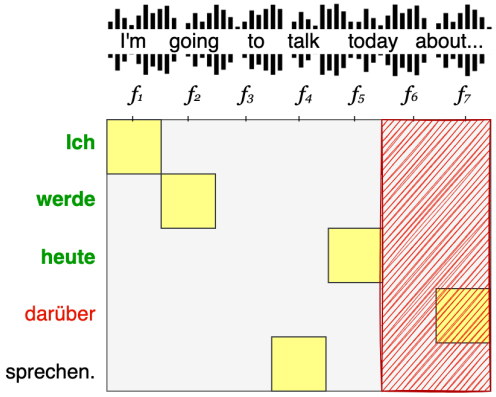
AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation

INTRODUCTION

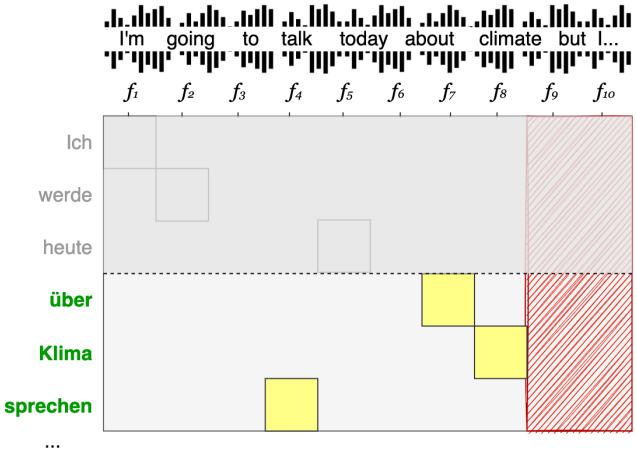
Simultaneous speech translation (SimulST) involves the generation, with minimal delay, of partial translations for an incrementally received input audio. In the quest for high-quality output and low latency, recent developments led to the advent of direct methods, which have been demonstrated to outperform the traditional cascaded (ASR + MT) pipelines in terms of both quality and latency (Anastasopoulos et al., 2022). Early works on direct SimulST require the training of several models which were optimized for different latency regimes (Ren et al., 2020; Ma et al., 2020b; Zeng et al., 2021), consequently resulting in high computational and maintenance costs. With the aim of reducing this computational burden, the use of offline-trained direct ST models for the simultaneous inference has been recently studied (Papi et al., 2022b) and is becoming popular (Liu et al., 2020b; Chen et al., 2021; Nguyen et al., 2021) due to its competitive performance compared to dedicated architectures specifically developed for SimulST (Anastasopoulos et al., 2022). Indeed, this approach enables an offline ST model to work in simultaneous by applying, only at inference time, a so-called *decision policy*, which is in charge of determining whether to emit a partial hypothesis or wait for more audio input. As a result, no specific adaptation is required either for the SimulST task or to achieve different latency regimes.

Along this line of research, we propose AlignAtt, a novel policy for SimulST that exploits the audio-translation alignments obtained from the attention weights of an offline-trained model to decide whether to emit or not a partial translation. Our policy is based on the idea that, if the candidate token is aligned with the last frames of the input audio, the information encoded can be insufficient to safely produce that token. The audio-translation alignments are automatically generated from the attention weights, whose representativeness has been extensively studied in linguistics-related tasks (Raganato and Tiedemann, 2018; Htut et al., 2019; Lamarre et al., 2022), including word-alignment in machine translation (Tang et al., 2018; Garg et al., 2019; Chen et al., 2020).

All in all, the contributions of our work are the following:



(1) The emission stops when "***Ich werde heute***" has been generated because the token "***darüber***" ("*about*") is aligned with an inaccessible frame (in striped red).



(2) After "*Ich werde heute*", also "***über Klima sprechen***" is emitted since no token is aligned with inaccessible frames.

Figure 3.17: Example of the AlignAtt policy with $f = 2$ at consecutive time steps t_1 (a) and t_2 (b).

- We present AlignAtt, a novel decision policy for SimulST that guides an offline-trained model during simultaneous inference by leveraging audio-translation alignments computed from the attention weights;
- We compare AlignAtt with popular and state-of-the-art policies that can be applied to offline-trained ST models, achieving the new state of the art on all the 8 languages of MuST-C v1.0 (Cattoni et al., 2021), with gains of 2 BLEU points and a latency reduction of 0.5-0.8s depending on the target languages;
- The code, the models, and the simultaneous outputs are published under Apache 2.0 Licence at: `https://github.com/hlt-mt/fbk-fairseq`.

ALIGNATT POLICY

AlignAtt is based on the source audio - target text alignment obtained through the attention scores of a Transformer-based model (Vaswani et al., 2017). In the Transformer, encoder-decoder (or cross) attention A_C is computed by applying the standard dot-product mechanism (Chan et al., 2016) as follows:

$$A_C(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where the matrices K (key) and V (value) are obtained from the encoder output and consequently depend on the input source $\mathbf{x}$, the matrix Q (query) is obtained from the output of the previous decoder layer (or from the previous output tokens in case of the first decoder layer), and consequently depends on the prediction $\mathbf{y}$, and d_k is a scaling factor. Cross attention can be hence expressed as a function of $\mathbf{x}$ and $\mathbf{y}$, obtaining $A_C(\mathbf{x}, \mathbf{y})$. Exploiting the cross attention $A_C(\mathbf{x}, \mathbf{y})$, the alignment vector $Align$ is computed by considering, for each token y_i of the prediction $\mathbf{y} = [y_1, ..., y_m]$, the index of the most attended frame (or encoder state) x_j of the source input $\mathbf{x} = [x_1, ..., x_n]$:

$$Align_i = \arg\max_j A_C(\mathbf{x}, y_i)$$

This means that, for every predicted token y_i, we have a unique aligned frame x_j of index $Align_i$.

Our policy (Figure 3.17) exploits the obtained alignment $Align$ to guide the model during inference by checking whether each token y_i attends to the last f frames or not. If this condition is verified, the emission is stopped, under the assumption that, if a token is aligned with the most recently received audio frames, the information they provide can be insufficient to generate that token (i.e. the system has to wait for additional audio input). Specifically, starting from the first token, we iterate over the prediction $\mathbf{y}$ and continue the emission until:

$$Align_i \notin \{n - f + 1, ..., n\}$$

which means that we stop the emission as soon as we find a token that mostly attends to one of the last f frames. Thus, f is the parameter that directly controls the latency of the model: smaller f values mean fewer frames to be considered inaccessible by the model, consequently implying a lower chance that our stopping condition is verified and, in turn, lower latency. The process is formalized in Algorithm 1.

Algorithm 1 AlignAtt

Require: $Align$, f, $\mathbf{y}$
  $i \leftarrow 1$
  $prediction \leftarrow [\ \ ]$
  $stop \leftarrow False$
  while $stop \neq True$ **do**
    if $Align_i \in \{n - f + 1, ..., n\}$ **then**
      $stop \leftarrow True$ ▷ inaccessible frame
    else
      $prediction \leftarrow prediction + y_i$
      $i \leftarrow i + 1$
    end if
  end while

Since in SimulST the source speech input $\mathbf{x}$ is incrementally received and its length n is increased at every time step t, applying the AlignAtt policy means applying Algorithm 1 at each timestep to emit (or not) the partial hypothesis until the input $\mathbf{x}(t)$ has been entirely received.

EXPERIMENTAL SETTINGS

3.2.3.3.1 Data

We train one model for each of the 8 languages of MuST-C v1.0 (Cattoni et al., 2021), namely English (en) to Dutch (nl), French (fr), German (de), Italian (it), Portuguese (pt), Romanian (ro), Russian (ru), and Spanish (es). We filter out segments longer than 30s from the training set to optimize GPU RAM consumption. We also apply sequence-level knowledge distillation (Kim and Rush, 2016) to increase the size of our training set and improve performance. To this aim, we employ NLLB 3.3B (Costa-jussà et al., 2022) as the MT model to translate the English transcripts of the training set into each of the 8 languages, and we use the automatic translations together with the gold ones during training. As a result, the final number of target sentences is twice the original one while the speech input remains unaltered. The performance of the NLLB 3.3B model on the MuST-C v1.0 test set is shown in Table 3.7.

Model	de	es	fr	it	nl	pt	ro	ru	Avg
NLLB	33.1	38.5	46.5	34.4	37.7	40.4	32.8	23.5	35.9

Table 3.7: BLEU results on all the language pairs of MuST-C v1.0 tst-COMMON of NLLB 3.3B model.

3.2.3.3.2 Architecture and Training Setup

The model is made of 12 Conformer (Gulati et al., 2020) encoder layers and 6 Transformer decoder layers, having 8 attention heads each. The embedding size is set to 512 and the feed-forward layers are composed of 2,048 neurons, with ∼115M parameters in total. The input is represented by 80 log Mel-filterbank audio features extracted every $10ms$ with a sample window of 25, and pre-processed by two 1D convolutional layers of striding 2 to reduce the input length by a factor of 4 (Wang et al., 2020a). Dropout is set to 0.1 for attention, feed-forward, and convolutional layers. The kernel size is 31 for both point- and depth-wise convolutions in the Conformer encoder. The SentencePiece-based (Sennrich et al., 2016) vocabulary size is 8,000 for translation and 5,000 for transcript. Adam optimizer with label-smoothed cross-entropy loss (smoothing factor 0.1) is used during training together with CTC loss (Graves et al., 2006) to compress audio input representation and speed-up inference time (Gaido et al., 2021a). Learning rate is set to $5 \cdot 10^{-3}$ with Noam scheduler and 25,000 warm-up steps. Utterance-level Cepstral Mean and Variance Normalization (CMVN) and SpecAugment (Park et al., 2019) are also applied during training. Trainings are performed on 2 NVIDIA A40 GPUs with 40GB RAM. We set 40k as the maximum number of tokens per mini-batch, update frequency 4, and 100,000 maximum updates (∼28 hours). Early stopping is applied during training if validation loss does not improve for 10 epochs. We use the bug-free implementation of fairseq-ST (Papi et al., 2023c).

3.2.3.3.3 Terms of Comparison

We conduct experimental comparisons with the other SimulST policies that can be applied to offline systems, thus policies that do not require training nor adaptation to be run, namely:

- **Local Agreement (LA)** (Liu et al., 2020b): the policy used by (Polák et al., 2022) to win the SimulST task at the IWSLT 2022 evaluation campaign (Anastasopoulos et al., 2022). With this policy, a partial hypothesis is generated each time a new speech segment is added as input, and it is emitted, entirely or partially, if the previously generated hypothesis is equal to the current one. We adapted the docker released by the authors to Fairseq-ST (Wang et al., 2020a). Different latency regimes are obtained by varying the speech segment length T_s.
- **Wait-k** (Ma et al., 2019a): the most popular policy originally published for simultaneous machine translation and then adapted to SimulST (Ren et al., 2020;

Zeng et al., 2021). It consists in waiting for a predefined number of words (k) before starting to alternate between writing a word and waiting for new output. We employ adaptive word detection guided by the CTC prediction to detect the number of words in the speech as in (Zeng et al., 2021; Papi et al., 2022b).

- **EDAtt** (Papi et al., 2023d): the only existing policy that exploits the attention mechanism to guide the inference. Contrary to our policy that computes audio-text alignments starting from the attention scores, in EDAtt the attention scores of the last λ frames are summed and a threshold α is used to trigger the emission. While α handles the latency, λ is a hyper-parameter that has to be empirically determined on the validation set. This represents the main flaw of this policy since, in theory, λ has to be estimated for each language. Here, we set $\lambda = 2$ following the authors' finding.

3.2.3.3.4 Inference and Evaluation

For inference, the input features are computed on the fly and Global CMVN normalization is applied as in (Ma et al., 2020b). We use the SimulEval tool (Ma et al., 2020a) to compare AlignAtt with the above policies. For the LA policy, we set $T_s = [10, 15, 20, 25, 30]$[20]; for the wait-k, we vary k in $[2, 3, 4, 5, 6, 7]$[21]; for EDAtt, we set $\alpha = [0.6, 0.4, 0.2, 0.1, 0.05, 0.03]$[22]; for AlignAtt, we vary f in $[2, 4, 6, 8, 10, 12, 14]$. Moreover, to be comparable with EDAtt, for our policy we extract the attention weights from the 4^{th} decoder layer and average across all the attention heads. All inferences are performed on a single NVIDIA TESLA K80 GPU with 12GB of RAM as in the IWSLT Simultaneous evaluation campaigns (Anastasopoulos et al., 2021, 2022). We use sacreBLEU (↑) (Post, 2018)[23] to evaluate translation quality and Length Adaptive Average Lagging (Papi et al., 2022a) – or LAAL (↓) – to measure latency.[24] As suggested by (Ma et al., 2020b), we report the computational-aware version of LAAL[25] that accounts for the real elapsed time instead of the ideal one, consequently providing a more realistic latency measure.

[20]Smaller values of T_s do not improve computational aware latency.

[21]We do not report results obtained with $k = 1$ since the translation quality highly degrades.

[22]These are the same values indicated by the authors of the policy.

[23]BLEU+case.mixed+smooth.exp+tok.13a+version.1.5.1

[24]Length Adaptive Average Lagging is a an improved speech version of Average Lagging (Ma et al., 2019a), which accounts for both longer and shorter predictions compared to the reference.

[25]We present all the results with $\text{LAAL}_{\text{max}} = 3.5s$.

RESULTS

In this section, we present the results of our offline systems trained for each language pair of MuST-C v1.0 to show their competitiveness compared to the systems published in the literature (Section 3.2.3.4.1) and the results of the AlignAtt policy compared to the other policies presented in Section 3.2.3.3.3 (Section 3.2.3.4.2).

3.2.3.4.1 Offline Results

To provide an upper bound to the simultaneous performance and show the competitiveness of our models, we present in Table 3.8 the offline results of the systems trained on all the language pairs of MuST-C v1.0 compared to systems published in literature that report results for all languages. As we can see, our offline systems outperform the others on all but 2 language pairs, en→{es, fr, it, nl, pt, ro}, achieving the new state of the art in terms of translation quality. BLEU gains are more evident for en→fr and en→it, for which we obtain improvements of about 1 BLEU point, while they amount to about 0.5 BLEU points for the other languages.

Concerning the other 2 languages (de, ru), our en→ru model achieves a similar result (18.4 vs 18.5 BLEU) with that obtained by the best model for that language (XSTNet (Ye et al., 2021)), with only a 0.1 BLEU drop. Moreover, our system reaches a slightly worse but competitive result for en→de (28.0 vs 28.7 BLEU) compared to STEMM (Fang et al., 2022), which instead makes use of a relevant amount of external speech data, and it also outperforms all the other systems for this language direction. On average, our approach stands out as the best one even if it does not involve the use of external speech data: it obtains an average of 29.4 BLEU across languages, which corresponds to 0.5 to 4.6 BLEU improvements compared to the published ST models.

3.2.3.4.2 Simultaneous Results

Having demonstrated the competitiveness of our offline models, we now apply the SimulST policies introduced in Section 3.2.3.3.3 to the same offline ST model for each language pair of MuST-C v1.0. Figure 3.18 shows the results in terms of latency-quality trade-off (i.e. LAAL (↓) - BLEU (↑) curves).

As we can see, our AlignAtt policy is the only policy, together with EDAtt, capable of reaching a latency lower or equal to $2s$ for all the 8 languages.[26] Specifically,

[26]The maximum acceptable latency limit is set between $2s$ and $3s$ from most works on simultaneous interpretation (Barik, 1975; Fantinuoli and Montecchio, 2022).

Model	Ext. Data		de	es	fr	it	nl	pt	ro	ru	Avg
	Speech	Text									
Fairseq-ST (Wang et al., 2020a)	-	-	22.7	27.2	32.9	22.7	27.3	28.1	21.9	15.3	24.8
ESPnet-ST (Inaguma et al., 2020)	-	-	22.9	28.0	32.8	23.8	27.4	28.0	21.9	15.8	25.1
Chimera (Han et al., 2021)	✓	✓	27.1	30.6	35.6	25.0	29.2	30.2	24.0	17.4	27.4
W-Transf. (Ye et al., 2021)	✓	-	23.6	28.4	34.6	24.0	29.0	29.6	22.4	14.4	25.8
XSTNet (Ye et al., 2021)	✓	✓	27.8	30.8	38.0	26.4	31.2	32.4	25.7	**18.5**	28.9
LNA-E,D (Li et al., 2021)	✓	✓	24.3	28.4	34.6	24.4	28.3	30.5	23.3	15.9	26.2
LightweightAdaptor (Le et al., 2021)	-	-	24.6	28.7	34.8	25.0	28.8	31.0	23.7	16.4	26.6
E2E-ST-TDA (Du et al., 2022)	✓	✓	25.4	29.6	36.1	25.1	29.6	31.1	23.9	16.4	27.2
STEMM (Fang et al., 2022)	✓	✓	**28.7**	31.0	37.4	25.8	30.5	31.7	24.5	17.8	28.4
ConST (Ye et al., 2022)	✓	-	25.7	30.4	36.8	26.3	30.6	32.0	24.8	17.3	28.0
ours	-	✓	28.0	**31.5**	**39.0**	**27.3**	**31.8**	**32.9**	**26.3**	18.4	**29.4**

Table 3.8: BLEU results on MuST-C v1.0 tst-COMMON. "Ext. Data" means that external data has been used for training: "Speech" means that either unlabelled or labelled additional speech data is used to train or initialize the model, "Text" means that either machine-translated or monolingual texts are used to train or initialize the model. "Avg" means the average over the 8 languages.

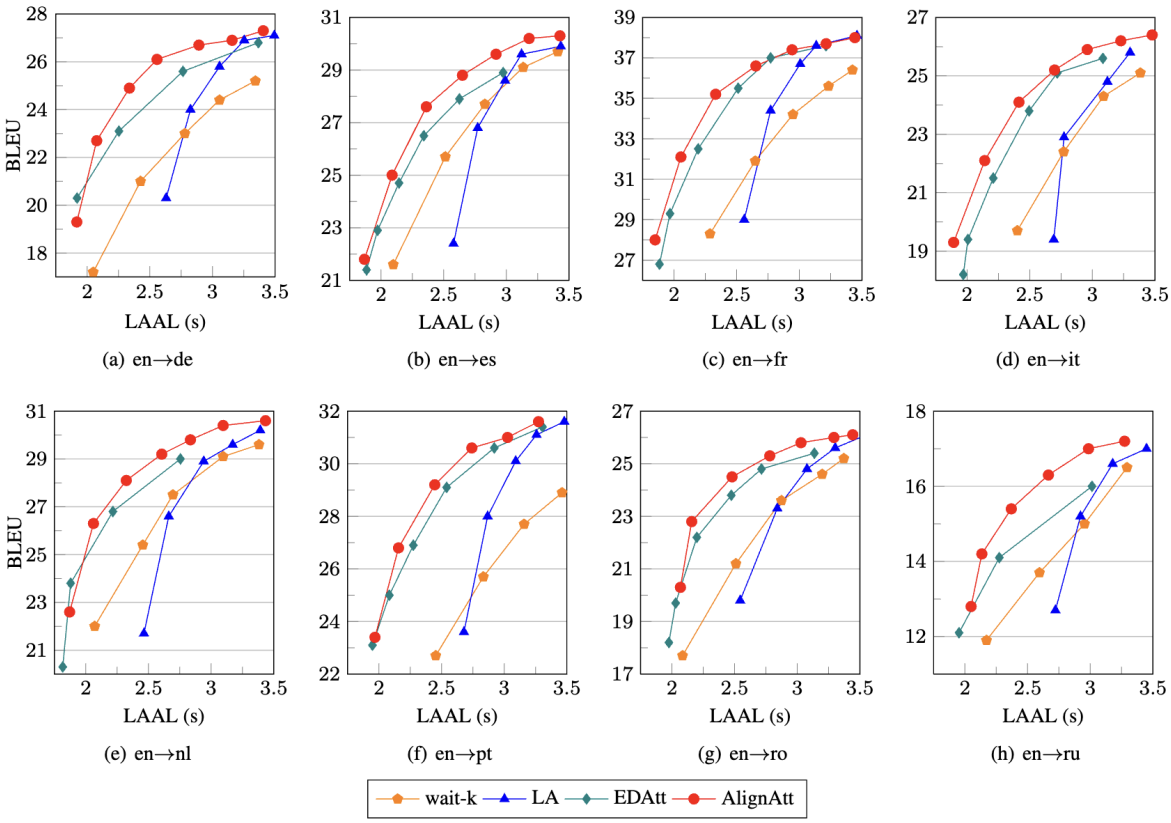


Figure 3.18: LAAL-BLEU curves for all the 8 language pairs of MuST-C tst-COMMON. AlignAtt is compared to the SimulST policy presented in Section 3.2.3.3.3. Latency (LAAL) is computationally aware and expressed in seconds (s).

LA curves start at around 2.5s or more for all the language pairs, even if they are able to achieve high translation quality towards 3.5s, with a 1.2 average drop in terms of BLEU across languages compared to the offline inference. Similarly, the wait-k curves start at around 2/2.5s but are not able to reach high translation quality even at high latency (LAAL approaching 3.5s), therefore scoring the worst results. Compared to these two policies, AlignAtt shows a LAAL reduction of up to 0.8s compared to LA and 0.5s compared to wait-k. Despite achieving lower latency as AlignAtt, the EDAtt policy achieves worse translation quality at almost every latency regime compared to our policy, with drops of up to 2 BLEU points across languages. These performance drops are particularly evident for en→de and en→ru, where the latter represents the most difficult language pair also in offline ST (it is the only language with less than 20 BLEU on Table 3.8). The evident differences in the AlignAtt and EDAtt policy behaviors, especially in terms of translation quality, prove that, despite both exploiting attention scores as a source of information, the decisions taken by the two policies are intrinsically different. Moreover, AlignAtt is the closest policy to achieving the offline results of Table 3.8, with less than 1.0 BLEU average drop versus 1.8 of EDAtt.

We can conclude that, on all the 8 languages of MuST-C v1.0, the AlignAtt policy achieves a lower latency compared to both wait-k and LA, and an improved translation quality compared to EDAtt, therefore representing the new state-of-the-art SimulST policy applicable to offline ST models.

CONCLUSIONS

We presented AlignAtt, a novel policy for SimulST that leverages the audio-translation alignments obtained from the cross-attention scores to guide an offline-trained ST model during simultaneous inference. Results on all 8 languages of MuST-C v1.0 showed the effectiveness of our policy compared to the existing ones, with gains of 2 BLEU and a latency reduction of 0.5-0.8s, achieving the new state of the art. Code, offline ST models, and simultaneous outputs are released open source to help the reproducibility of our work.

Chapter 4

Automatic Subtitling

4.1 Background

Automatic subtitling is the task in which the content of audio-visual resources (e.g., YouTube videos, TV series, movies, and video lectures) has to be transcribed in the source language (*intralingual* subtitles) or translated into another language (*interlingual* subtitles), and organized in a subtitle block displaying the text (element 1 in Figure 4.1) associated with timestamp information indicating the start and the end time of its on-screen duration (element 2 in Figure 4.1). Throughout this PhD thesis, I focused

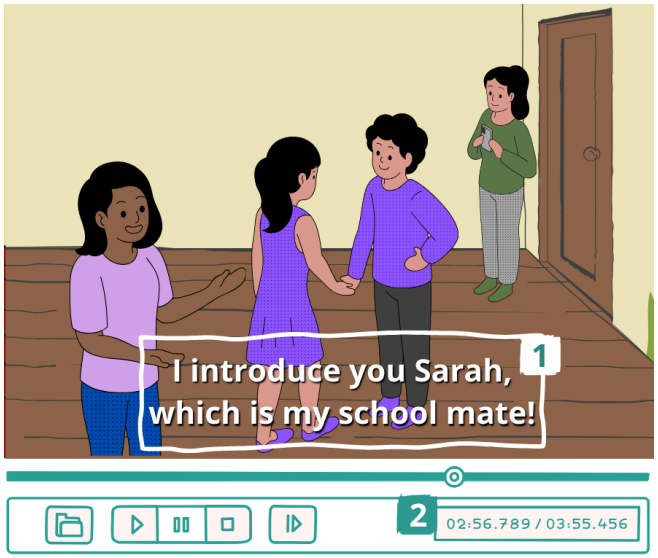


Figure 4.1: Example of a subtitle composed of a block of text (1), and the corresponding timestamp (2).

on automatizing interlingual subtitling (hereinafter, only subtitling), framing it as an

application of ST.

Differently from standard ST, in automatic subtitling, the generated text has to comply with multiple requirements related to its length, format, and the time it is displayed on the screen (Cintas and Remael, 2021). These particular requirements, which naturally vary depending on the nature of the video content and the intended target language and audience, are fundamentally driven by the need to minimize the cognitive load placed on viewers, optimizing comprehension and maintaining the audience engagement (Perego, 2008; Szarkowska and Gerber-Morón, 2018). This frequently results in a process of condensation applied to the original spoken content, aimed at reducing the time viewers spend reading subtitles, thereby allowing them to dedicate more time to the actual video content (Burnham et al., 2008; Szarkowska et al., 2016). In essence, automatic subtitling seeks to strike a balance between improving comprehension and sustaining viewer engagement, all while considering the particular language preferences of the target audience.

Despite the continuous growth of websites and streaming platforms such as YouTube and Netflix,[1] along with the consequent dramatic increase in the amount of audiovisual content available online that necessitates subtitles,[2] when I started my PhD journey there had been limited research dedicated to the advancement of automated subtitling tools (Álvarez et al., 2015; Vitikainen and Koponen, 2021).

Initial efforts to (semi-)automate the subtitling process have primarily involved the deployment of cascade systems (Piperidis et al., 2004; Melero et al., 2006; Matusov et al., 2019; Koponen et al., 2020; Bojar et al., 2021) comprising:

- an ASR model, which transcribes the uttered speech by also producing the timestamp information for each transcribed word;
- a subtitle segmenter, which segments the transcriptions into timed blocks and lines;
- an MT model, which translates the subtitle-segmented transcriptions into the desired target language.

In most of the works on automatic subtitling, a significant focus had been directed towards adapting the MT module specifically for the task, with a prominent objective being the generation of more concise and compressed textual content. This adaptation

[1]For instance, Netflix almost doubled its revenues in 2022 (31.62B USD) compared to 2018 (`https://www.statista.com/outlook/dmo/digital-media/video-on-demand/video-streaming-svod`).

[2]Netflix's original production hours have increased from just under 1,200 in 2017 to just over 3,500 in 2022 (`https://omdia.tech.informa.com/OM029224/Online-Original-Production--2022`).

was realized through various techniques, such as statistical approaches trained on subtitling corpora (Volk et al., 2010; Etchegoyhen et al., 2014; Bywood et al., 2013), as well as the development of specialized decoding solutions for both statistical (Aziz et al., 2012) and neural models (Matusov et al., 2019). More recently, the research has been concentrated on controlling the length of MT model outputs to satisfy isometric requirements between source transcripts and target translations (Lakew et al., 2019; Matusov et al., 2020; Lakew et al., 2021, 2022). Furthermore, studies conducted by Öktem et al. (2019); Federico et al. (2020); Virkar et al. (2021); Tam et al. (2022); Effendi et al. (2022) have underscored the utility of incorporating prosodic cues, such as pauses, in determining subtitle boundaries. Along this line of research, Karakanta et al. (2020a, 2021a) proposed the first direct ST system able both to translate the spoken content and segment the text into subtitles, confirming with their results that the ability of direct ST systems to leverage prosody has particular importance for subtitle segmentation. Despite the promising advancements, their study has two major drawbacks:

1. it is limited to only one domain, TED talks,[3] since the model is trained on the only existing corpus comprising both audio and subtitles available online, MuST-Cinema (Karakanta et al., 2020b);

2. it only covers the translation and segmentation into subtitles, completely neglecting the timestamp generation, which is left to external components.

Given the aforementioned limitations, during my PhD studies on automatic subtitling, I identified two main research questions: **1) Is there a way to exploit the ST corpora that already exist (and cover many domains other than TED talks) for automatic subtitling?** and **2) Is it possible to exploit a direct ST model for producing full subtitles (translated texts with their corresponding timestamps)?**

Before delving into my work to answer these questions, in the following, I will explain in detail two fundamental aspects of subtitling: the **subtitling guidelines** comprising all the requirements defining a good subtitle, and the **evaluation metrics** used to evaluate the quality of the segmentation, the content of the subtitle itself, and its temporal synchronization.

[3] `https://www.ted.com/`

4.1.1 Subtitling Guidelines

Subtitles are short pieces of timed text, generally displayed at the bottom of the screen, which describe, transcribe, or translate the dialogue or narrative (Figure 4.2).

Figure 4.2: Example of a subtitled image. See `https://commons.wikimedia.org/wiki/File:Example_of_subtitles_(Charade,_1963).jpg` for licence.

A subtitle is composed of two elements: the text, shown into "blocks", and the corresponding start and end display time – or timestamps. The segmentation in blocks is generally indicated by a specific marker `<eob>`, and the new line within each block is indicated by another marker `<eol>`, as shown in Figure 4.3.

164
00:08:57,020 –> 00:08:58,476 164
I wanted to challenge the idea

165
00:08:58,500 –> 00:09:02,060
that design is but a tool
to create function and beauty.

I wanted to challenge the idea <eob> that design is but a tool <eol> to create function and beauty. <eob>

Figure 4.3: Example of a subtitle in the widely used subtitle format SubRip (srt): The first element denotes the sequential number, the second contains the start and end timestamps, and the third presents the subtitle block with its textual content divided into lines. Below, is the subtitle content representation in blocks and lines, denoted by `<eol>` and `<eob>` markers.

Depending on the subtitle provider and the audiovisual content, different requirements have to be respected concerning both spatial and temporal constraints or, in other words, the text space and its temporal synchronization. These constraints typically consist in:

1. using at most two lines per block;

2. keeping linguistic units (e.g. noun and verb phrases) together in the same line;

3. not exceeding a pre-defined number of characters per line (CPL), spaces included;

4. not exceeding a pre-defined reading speed for each block, measured in the number of characters per second (CPS).

In particular, point 1. is to avoid occupying too much space on the screen so as not to interfere with what the video intends to show, points 2. and 4. aim to make the content easier and faster to read by requiring low cognitive effort, and point 3. is to facilitate on-screen reading in terms of physical effort, avoiding saccades (i.e., quick, simultaneous movements of both eyes between two or more phases of fixation in the same direction)[4]. Moreover, a fundamental requirement is the need for subtitles to be synchronized with the corresponding audiovisual content.

While a typical value used as the maximum CPL threshold is 42 for most Latin languages,[5] there is no agreement on the maximum CPS allowed. For instance, Netflix guidelines[6] allow up to 17 CPS for adults and 15 for children programs, The Walt Disney Studios[7] up to 20 CPS for adults and 17 for children, TED guidelines[8] up to 21 CPS, and Amara guidelines[9] up to 25 CPS. In addition, some subtitle providers, such as Netflix and The Walt Disney Studios, impose some constraints on the minimum duration of each subtitle, which is set to about 800 milliseconds, and on the maximum duration, which is set to 7 seconds, as well as on the minimum gap between two subtitles of 70-80 milliseconds.

Concerning text segmentation into subtitles, many subtitle providers, such as TED and Netflix, specifically instruct subtitlers on how to perform it, providing examples of how to balance the textual content in the subtitles, which are the typical words that cannot be followed by a line break (e.g., after “a”, “an”, “the”, “which”, “that”, and

[4] https://en.wikipedia.org/wiki/Saccade

[5] https://www.ted.com/participate/translate/subtitling-tips

[6] https://partnerhelp.netflixstudios.com/hc/en-us/articles/219375728-Timed-Text-Style-Guide-Subtitle-Templates

[7] Disney Digital Supply Chain Subtitle and Closed Captioning Style Guide of 2019 (https://disneymasteringspecs.s3.amazonaws.com/Disney_Digital_Supply_Chain_Subtitleand_CC_Style_Guide_1_1_1_2022_06_06_77ae3ac064.pdf).

[8] https://www.ted.com/participate/translate/subtitling-tips

[9] https://blog.amara.org/2020/10/22/create-quality-subtitles-in-a-few-simple-steps/

"who"), what type of linguistic units cannot be split (e.g., person names, nouns with their adjectives, and verbs with their subjects), just to mention few.[10]

As already mentioned in Section 4.1, to convey the meaning of the audiovisual product while adhering to time and space constraints, in some domains and scenarios, subtitles require compression or condensation (Kruger, 2001; Gottlieb, 2004; Aziz et al., 2012; Liu et al., 2020a; Buet and Yvon, 2021). For instance, TED guidelines suggest compressing subtitles over 21 CPS while trying to preserve as much meaning as possible. However, being the compression task less structured, there are no specific suggestions about how to correctly compress subtitles but only recommendations on how to do it without changing the meaning of the textual content or what are the cases in which a subtitle must not be compressed.[11]

Lastly, there are some guidelines that are language-specific, such as for French, Dutch, and Chinese, containing ad-hoc recommendations for each target language. For instance, the European Association for Studies in Screen Translation collects many guidelines spanning from French to German to Chinese languages,[12] similar to the AudioVisual Translators Europe for some European languages.[13]

Building upon the discussion about subtitling guidelines, it is evident that automatic subtitling is an inherently multifaceted task, requiring the generation of subtitles that adhere to a multitude of constraints. This adherence can result in a diverse set of subtitles, each variation being equally acceptable based on the established criteria. However, this inherent diversity introduces challenges in the evaluation process, as capturing the varied dimensions of subtitling quality becomes challenging. The difficulty arises from the absence of comprehensive and accurate metrics tailored for this intricate task. Existing metrics, if available, often specialize in different aspects of subtitling, lacking a holistic view. Consequently, as we will see in the next section, assessing the overall quality of automatic subtitling becomes a complex undertaking, requiring an evaluation framework capable of addressing the diverse dimensions of subtitled content accurately.

[10] For more examples, refer to the TED guidelines on lines breaking (`https://translations.ted.com/How_to_break_lines`).

[11] For more details, refer to the TED guidelines on subtitles compression (`https://translations.ted.com/How_to_Compress_Subtitles`).

[12] `https://esist.org/resources/avt-guidelines-and-policies/#interlingual_subtitling`

[13] `https://avteurope.eu/what-is-av-translation/standards/`

4.1.2 Evaluation

As already discussed in Section 1.2.2, subtitles have to satisfy specific constraints in terms of quality, space, and time. For evaluating translation quality, the **BLEU** metric (Papineni et al., 2002; Post, 2018) is commonly employed, similarly to SimulST (Chapter 3) and generic ST (Chapter 2), and computed over the texts without `<eob>` and `<eol>` markersz.

For the segmentation quality, instead, several metrics have been proposed over time, starting from 1999. In (Beeferman et al., 1999), the authors proposed to assign penalties for each moving window if subtitle ends are detected to be in different segments between reference and hypothesis. In (Pevzner and Hearst, 2002), the **WindowDiff** metric was proposed, consisting of assigning a penalty if the number of boundaries (`<eob>` or `<eol>`) in each window is different for reference and hypothesis. Álvarez et al. (2016) proposed precision, recall, and F1 (the harmonic mean of precision and recall). Precision was defined as the proportion of boundaries in the hypothesis that agree with the reference boundaries over the total number of hypothesis boundaries, while recall was defined as the number of correct boundaries divided by the reference boundaries. In following studies, edit distance-based metrics were proposed: **Segmentation similarity** (Fournier and Inkpen, 2012), which computes the proportion of boundaries that are not transformed when comparing segmentations using edit distance as a penalty function, and **Boundary similarity** (Fournier, 2013), which is an adaptation of segment similarity, where different weights are applied for each edit type. Building upon the concept of edit distance, Karakanta et al. (2020a) introduced $\mathbf{TER_{br}}$. In this metric, all words, excluding `<eob>` and `<eol>`, within each hypothesis-reference pair are masked, and TER (Snover et al., 2006) is subsequently computed over the masked sequences. Exploiting BLEU, the authors also proposed $\mathbf{BLEU_{br}}$, where BLEU is computed on text containing subtitle boundaries (`<eob>` and `<eol>`) as special symbols.

The major drawback of all these metrics, except for $\mathbf{TER_{br}}$ and $\mathbf{BLEU_{br}}$, is that they cannot be computed on imperfect texts i.e., hypotheses whose text does not match that of the reference, making their use impractical for evaluating automatically-translated subtitles.

More recently, Karakanta et al. (2022) proposed **Sigma** (S), which is based on BLEU_{br} and is formulated as:

$$S = \frac{BLEU_{br}}{BLEU^{+}_{br}}$$

where $BLEU^{+}_{br}$ is the upper bound of BLEU_{br} and is obtained by computing the standard

BLEU score on the translated text without subtitle boundaries. Sigma values close to 100 should represent a good segmentation, while values close to 0 indicate a bad segmentation, irrespective of the value of BLEU. Through comparisons with all the aforementioned metrics, Sigma proved its effectiveness both with perfect and imperfect texts, resulting in the best metric for measuring segmentation quality.

Regarding the evaluation of timestamp quality of a generated subtitle, a metric exclusively measuring this aspect does not exist yet. Instead, two metrics have been proposed in the literature to measure the overall quality of the produced subtitles including time: **t-BLEU** (Cherry et al., 2021), and **SubER** (Wilken et al., 2022).

The first metric, Timed BLEU or t-BLEU, is based on computing the standard BLEU score over temporally aligned target-reference segment pairs. The temporal alignment is realized by assigning a timestamp to each token in the target by linearly interpolating the subtitle timings, and then by assigning the target to the reference subtitle segments based on temporal overlap. However, a bad estimation of a target word timestamp can result in its misalignment with a segment without a corresponding reference word, or the word can even be dropped from the hypothesis if it does not fall into any reference segment.

To overcome this major drawback, SubER was introduced, which exploits the Levenshtein distance (Levenshtein, 1966) and is computed as:

$$\text{SubER} = \frac{\#\text{ word edits} + \#\text{ break edits} + \#\text{ shifts}}{\#\text{ reference words} + \#\text{ reference breaks}}$$

where "#" indicates "number of", "word edits" are insertions, deletions, and substitutions (allowed only if the hypothesis and reference words are from subtitles that overlap in time), "break edits" are insertions, deletions, and substitutions of `<eob>` and `<eol>` breaks (allowed between breaks not between a word and a break and within the time overlap), and "shifts" are movements of one or more adjacent hypothesis words and/or breaks to a position of the matching reference phrase, which is allowed only if the words overlap in time with the reference. The authors compared SubER with the other subtitling metrics, including t-BLEU, and showed that their metric better correlates with human judgment, thus resulting in the best metric for automatic subtitling.

Regarding the conformity constraints mentioned in Section 4.1.1, two main measures can be computed from subtitles: characters-per-line (CPL) and characters-per-second (CPS). The **CPL** is calculated by computing the percentage of subtitles featuring a maximum length of 42 characters, which is the maximum limit according to standard

subtitling guidelines (Section 4.1.1):

$$CPL(\%) = \sum_{i=1,\dots,N} \frac{\text{len-conform}_i}{N} * 100$$

where

$$\text{len-conform}_i = \begin{cases} 1, & \text{if} \quad \text{len}(subtitle_i) \leq 42 \\ 0, & \text{otherwise} \end{cases}$$

and the length of the subtitles (len(*subtitle*)) is computed as the number of characters in the subtitle block, excluding `<eob>` and `<eol>` but including spaces, and N is the total number of subtitle blocks.

The **CPS** is calculated by computing the percentage of subtitles featuring a maximum of 21 characters per second, which is the maximum limit according to TED guidelines, similar to The Walt Disney guidelines and in between Netflix and Amara guidelines (Section 4.1.1):

$$CPS(\%) = \sum_{i=1,\dots,N} \frac{\text{time-conform}_i}{N} * 100$$

where

$$\text{time-conform}_i = \begin{cases} 1, & \text{if} \quad \frac{\text{len}(subtitle_i)}{\text{time}(subtitle_i)} \leq 21 \\ 0, & \text{otherwise} \end{cases}$$

and the time duration of the subtitle blocks (time(*subtitle*)) is obtained by the associated timestamp (end time - start time).

In practice, these two metrics, CPL and CPS, evaluate the percentage of subtitles that conform to, respectively, the length and time constraints[14] and are very important indicators of what is the impact of the subtitles on the screen and, consequently, on users' experience.

4.2 Selected Contributions

In the context of training automatic subtitling (AS) systems, the data scarcity problem, which also affects standard ST (Section 2.3), is exacerbated by the absence of datasets containing subtitle-like texts with `<eob>` and `<eol>`. To tackle this problem, during the first phase of this PhD, I focused on the generation of synthetic data to develop AS

[14] 42 and 21 are the values (arbitrary) that have become established in research on the topic, and that I too have adopted in my work on subtitling. However, nothing is preventing the use of other limits depending on the context and usage scenario.

systems. The fundamental question was: **Is there a way to exploit the already existing ST corpora, which span multiple domains, for automatic subtitling?**

The primary goal was to devise a method for automatically segmenting translated texts into subtitles accurately without compromising the final AS model performance. With this objective in mind, in my first selected contribution (**PAPER #1**: "*Dodging the Data Bottleneck: Automatic Subtitling with Automatically Segmented ST Corpora*", Section 4.2.1), I compared various models for the segmentation task and proposed a multilingual multimodal segmenter capable of taking both audio and text as input to perform segmentation. This approach was motivated by previous work in subtitling (Section 4.1), which proved the importance of speech cues, such as pauses, for text segmentation into subtitles (Öktem et al., 2019; Federico et al., 2020; Virkar et al., 2021; Tam et al., 2022; Effendi et al., 2022). The main advantage of starting the generation of the synthetic subtitled data from ST corpora is that both audio and text are available and can be exploited to produce a better segmentation.

In practice, the proposed automatic segmenter processes plain texts (with corresponding speeches for the multimodal segmenter) and adds subtitle markers `<eob>` and `<eol>` without altering the original texts. A direct ST system is then trained on these audio-automatically segmented text pairs to produce the translations with `<eob>` and `<eol>` directly from the audio in an end-to-end fashion. Therefore, this model would allow the exploitation of any existing ST corpus for the subtitling task.

Through extensive comparisons with textual segmenters (i.e., segmenters that take only plain texts as input) trained on the same data and the very large OpenSubtitles (Lison et al., 2018)), I showed that the proposed multilingual multimodal segmenter not only yields more accurate segmentations while maintaining high length conformity (CPL%) but also generalizes better to unseen languages (i.e., language pairs that have not been used to train the model). This applies to both similar languages (for example, a model trained in Italian and used in zero-shot for Spanish, another Latin language) but also to different languages (for example, a model trained in Italian and used in zero-shot for Dutch, a Germanic language). Moreover, multilingualism, that is exposing the model to multiple languages during training, can further enhance the performance of the segmenter compared to training separate ad-hoc segmenters for each language.

With this automatic segmentation tool, it became possible to train and compare direct ST systems across various domains and data scenarios. Nevertheless, a fundamental component was still missing from the direct ST systems at that time: the timestamp estimation. Filling this gap was the objective of my subsequent research in automatic subtitling, where I aimed to answer the question: **Is it possible to exploit a direct ST**

model for producing "full" subtitles also including timestamp information?

Working on this problem, in my second selected contribution (**PAPER #2**:"*Direct Speech Translation for Automatic Subtitling*"), I proposed the first full-automatic subtitling system that leverages a direct ST model to produce full-automatic subtitles. The direct ST model was trained on translations with `<eob>` and `<eol>` markers (either automatically or manually assigned across different experimental scenarios) using a standard cross-entropy loss in combination with an auxiliary CTC loss (Graves et al., 2006) trained on segmented transcripts. To retrieve timestamp information, I exploited the CTC predictions obtained through the auxiliary loss, which directly maps the time frames with corresponding target tokens. Specifically, I used the CTC segmentation algorithm (Kürzinger et al., 2020) that, starting from the subtitle-segmented transcription, is able to obtain the time information from the corresponding subtitle blocks. The resulting timestamps are then projected into the target, providing the timestamp-subtitle pairs necessary to produce "full" timed subtitles.

This AS system was trained under both constrained data conditions (specifically on all the language pairs of MuST-Cinema (Karakanta et al., 2020b)), and unconstrained data conditions (using data available for the IWSLT Evaluation Campaign on Automatic Subtitling[15]). The performance was evaluated against a cascade architecture trained on the same data, as well as five different production tools.

To further enrich the evaluation, two new test sets were also proposed in the paper. One set consisted of short videos from the European Commission[16], covering various topics (e.g., inclusivity, and environmental problems), and featuring background music and multiple speakers. The other set included interviews with mostly non-native speakers from the European Parliament[17], with highly compressed subtitles. These test sets provided an opportunity to benchmark AS systems moving a step further towards diverse domains that pose additional challenges compared to TED talks, including the presence of background noise, multiple speakers, and the need to generate subtitles with a higher level of compression.

Through comprehensive comparisons with both cascade solutions and production tools, the proposed system not only emerged as a viable alternative to existing approaches but also outperformed them in terms of SubER, despite not being initially designed for production purposes, achieving the new state-of-the-art in automatic subtitling.

[15] `https://iwslt.org/2023/subtitling`
[16] `https://commission.europa.eu`
[17] `https://www.europarl.europa.eu`

SUMMARY

My research journey in the field of automatic subtitling has been driven by the goal of advancing the adoption of direct ST systems for this specific task. Early in my research, I recognized a significant challenge posed by data scarcity in the context of automatic subtitling. To address this challenge, I proposed a method for synthetically generating AS corpora. This approach allowed me to bridge the gap between the limited availability of AS data and the growing need for automatic subtitling tools.

Once this initial obstacle was successfully overcome, I shifted my focus toward a more comprehensive and forward-looking study. The central theme of this phase of my research was to develop direct ST systems tailored for "full" automatic subtitling. An in-depth analysis of the proposed solution covering diverse domains and data scenarios enabled a thorough understanding of the model adaptability and performance across a wide spectrum of conditions. Additionally, performance comparison against state-of-the-art methods and production tools revealed that our proposed solution can achieve comparable results and sometimes even surpass them.

By taking this multifaceted approach, my research has not only introduced groundbreaking solutions but also laid the foundation for future advancements in automatic subtitling. These contributions collectively reinforce the idea that direct ST systems hold the potential to revolutionize automatic subtitling technology, providing efficient and high-quality solutions to meet the demands of an ever-evolving media and communication landscape.

In the subsequent sections (Sections 4.2.1, and 4.2.2), these major contributions are presented, as they best represent my journey through automatic subtitling:

- **PAPER #1 (Papi et al., 2022c):**
  - **Publication details**:
    - ***Title***: Dodging the Data Bottleneck: Automatic Subtitling with Automatically Segmented ST Corpora
    - ***Authors***: Sara Papi, Alina Karakanta, Matteo Negri, Marco Turchi
    - ***Venue***: AACL 2022
  - **Research Question(s)**: *Can we cope with the data scarcity issue of automatic subtitling? Can we automatically but accurately segment the existing ST corpora into subtitles?*
  - **Main Contribution(s)/Finding(s)**: Automatic segmentation into subtitles can be effectively achieved by a multimodal segmenter, which exploits both

audio and texts in order to find the best segmentation points, and the subtitle-segmented data can be used to train a SubST model with no significant drop in the performance compared with gold segmentation.

- **PAPER #2 (Papi et al., 2023a):**
  - **Publication details**:
    - ***Title***: Direct Speech Translation for Automatic Subtitling
    - ***Authors***: Sara Papi, Marco Gaido, Alina Karakanta, Mauro Cettolo, Matteo Negri, Marco Turchi
    - ***Venue***: Transactions of ACL (TACL) 2023
  - **Research Question(s)**: *Can we leverage a direct ST model to produce the full subtitles (comprising both texts segmented into subtitles and their timestamp)?*
  - **Main Contribution(s)/Finding(s)**: The first study on exploring direct ST models for full automatic subtitling (text segmented into subtitles with their corresponding timestamp that must adhere to spatio-temporal constraints), which shows that these models can effectively produce full subtitles and are also competitive with production tools.

4.2.1 PAPER #1

Dodging the Data Bottleneck: Automatic Subtitling with Automatically Segmented ST Corpora

INTRODUCTION

Massive amounts of audiovisual content are available online, and this abundance is accelerating with the spread of online communication during the COVID-19 pandemic. The increased production of pre-recorded lectures, presentations, tutorials and other audiovisual products raises an unprecedented demand for subtitles in order to facilitate comprehension and inclusion of people without access to the source language speech. To keep up with such a demand, automatic solutions are seen as valuable support to the limited human workforce of trained professional subtitlers available worldwide Tardel (2020). Attempts to automatize subtitling have focused on Machine Translation for translating human- or automatically-generated source language subtitles Volk et al. (2010); Etchegoyhen et al. (2014); Matusov et al. (2019); Koponen et al. (2020). Recently, direct ST systems Bérard et al. (2016); Weiss et al. (2017) have been shown to achieve high performance while generating the translation in the target language without intermediate transcription steps. For automatic subtitling, Karakanta et al. (2020a) suggested that, by directly generating target language subtitles from the audio (i.e. predicting subtitle breaks together with the translation), the model can improve subtitle segmentation by exploiting additional information like pauses and prosody. However, the scarcity of SubST corpora makes it hard to build competitive systems for automatic subtitling, especially if no corpus is available for specific languages/domains.

One solution to the SubST data bottleneck could be leveraging ST corpora by inserting subtitle breaks on their target side. Automatic segmentation of a text into subtitles is normally implemented with rule-based approaches and heuristics, e.g. a break is inserted before a certain length limit is reached. More involved algorithms (SVM, CRF, seq2seq) predict breaks using a segmenter model trained on subtitling data for a particular language Álvarez et al. (2016, 2017); Karakanta et al. (2020c). Still, the performance of these models relies on high-quality segmentation annotations for each language, which web-crawled subtitling corpora like OpenSubtitles Lison et al. (2018) rarely contain.

In this work, we address the scarcity of SubST corpora by developing a multimodal

segmenter able to automatically annotate existing ST corpora with subtitle breaks in a zero-shot fashion. Specifically, our segmenter exploits, for the first time in this scenario, the source language audio (here: en) and segmented target text already available in a few languages (here: de, en, fr, it). Its key strength is the ability to segment not only target languages for which high-quality segmented data is available but also unseen languages having some degree of similarity with those covered by the original ST resource(s). This opens up the possibility of automatically obtaining synthetic SubST training data for previously not available languages. Along this direction, our zero-shot segmentation results in two unseen languages (es, nl) show that training a SubST system on automatically segmented data leads to comparable performance compared to using a gold, manually-segmented corpus.

METHOD

Our approach for leveraging ST corpora for SubST is summarized as follows: *i)* initially, we train various segmenters on available human-segmented subtitling data to identify the most effective one; *ii)* next, we apply the selected segmenter in a *zero-shot* manner (i.e. without fine-tuning or adaptation) to insert subtitle breaks into unsegmented texts of *unseen* languages; *iii)* then, we pair the automatically annotated texts with their corresponding audio to create a synthetic parallel SubST corpus; *iv)* lastly, we train a SubST model on the synthetic corpus.

We evaluate the effectiveness of our approach on two language pairs (en-es, en-nl) by conducting a comparative analysis between SubST models trained on synthetic data and those trained on the original gold data.

Segmenter. We adopt the general segmentation approach introduced in Karakanta et al. (2020b), which employs a sequence-to-sequence *Textual segmenter*, trained on pairs of unsegmented-segmented text, to insert subtitle breaks into unsegmented text.

To enhance the quality of segmentation, we extend this approach in two ways. Our first extension involves multimodal training. Given that speech-related phenomena such as pauses and silences have a significant impact on subtitle structure (Carroll and Ivarsson, 1998), we expect that incorporating information from the speech modality could enhance segmentation quality. To investigate this hypothesis, we extend the textual segmenter with a multimodal architecture (Sulubacak et al., 2020), capable of receiving input from different modalities, specifically audio and text. [18]

[18]Images and videos with subtitling material are often protected by copyright and thus not publicly available. Improving the segmenter with data from the visual modality is thus left to future work, contingent on the availability of such resources.

Our *Multimodal segmenter* is constructed using a dual-encoder architecture: one for processing text (with the same structure as the textual segmenter) and one for processing audio. We combine the encoder states obtained by the two encoders using parallel cross-attention (Bawden et al., 2018),[19] as it has demonstrated to be effective in both speech and machine translation (Kim et al., 2019; Gaido et al., 2020a). The parallel attention mechanism (Figure 4.4) operates by attending to the same intermediate representation (the decoder self-attention); then, the cross-attention from the audio encoder and the text encoder are summed together and fed to the feed-forward layer.

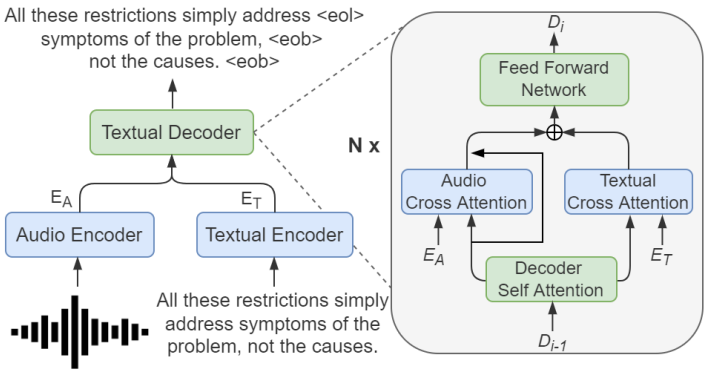


Figure 4.4: Parallel Multimodal segmenter architecture.

Given that subtitling constraints remain consistent across multiple languages, our second extension is to learn segmentation multilingually. To this aim, we employ established techniques commonly utilized in MT and ST, respectively: for the textual segmented, we combine samples from multiple languages within the same training step (Ott et al., 2018); for the multimodal segmenter, we introduce a language prefix token to the target text (Inaguma et al., 2019). This multilingual training approach, similar to that employed in MT (Ha et al., 2016), has been demonstrated to boost performance (Wang et al., 2020a) while maintaining only one model for multiple languages.

EXPERIMENTAL SETTINGS

Data. To train our textual and multimodal segmenters, we use en→{de, fr, it} sections of MuST-Cinema (Karakanta et al., 2020b),[20] the only publicly available SubST dataset. Each section contains paired audio utterances, English transcripts, and translations in the corresponding language, where both sides of the text are built from subtitles created

[19]We also tried sequential cross-attention (Zhang et al., 2018) but we do not report these results since they are slightly worse compared to parallel cross-attention.

[20]`https://ict.fbk.eu/must-cinema/` - License: CC BY-NC-ND 4.0

by humans. For French (275K sentences), German (229K sentences) and Italian (253K sentences), we collect the segmented translations of the corresponding MuST-Cinema sections. For English, we concatenate the segmented transcripts of the previous three sections (757K sentences). For each language (de, en, fr, it), the training data for the segmenter consists of unsegmented texts and, in the case of the multimodal segmenter, audio as the source input, and segmented texts (subtitles) as the target. Using the corpus notation, subtitle breaks are defined as: *block break* `<eob>`, which marks the end of the current subtitle displayed on the screen, and *line break* `<eol>`, which splits consecutive lines inside the same block. For unsegmented texts, `<eob>` and `<eol>` are removed.

To test the segmenters in zero-shot conditions and train our SubST models, we select two target languages also contained in MuST-Cinema:[21] Dutch (an SOV – Subject-Verb-Object – language) and Spanish (SVO).

Baselines. We compare the performance of the segmenters with two baselines. One is a rule-based method (*Count Chars*) where a break is inserted before a 42-character limit. This is the simplest method to always produce length-conforming subtitles and serves as a lower bound for segmentation performance. Our second baseline (*Supervised*) is a neural textual segmenter trained on OpenSubtitles, the largest collection of publicly available textual subtitling data, for the respective language (es, nl). Although OpenSubtitles is available for a variety of languages, it has some limitations: it does not contain audio, the subtitle and segmentation quality varies since subtitles are often machine-translated or created by non-professionals, and line breaks were lost when pre-processing the subtitles to create the corpus. These limitations may have a detrimental effect on the quality of segmenters trained on this data (Karakanta et al., 2019).

Architectures and Training Settings. The *Textual* segmenter is a Transformer-based (Vaswani et al., 2017) architecture consisting of 3 encoder layers and 3 decoder layers. We set the hyper-parameters as in the fairseq (Ott et al., 2019) multilingual translation task, both for the mono- and multilingual textual segmenters. For the multilingual model, a mini-batch for each language direction (here: 4) is built and the model weights are updated after each mini-batch, a mechanism already present in fairseq Multilingual Machine Translation (Ott et al., 2019).

The *Multimodal* segmenter is an extension of the textual segmenter encoder-decoder structure with an additional speech encoder composed of 12 Transformer encoder layers as in the original speech-to-text task (Wang et al., 2020a) but with the addition of a CTC

[21]Though present in MuST-Cinema, es and nl data are only used for testing purposes so as to simulate the zero-shot conditions required to select the best segmenter and evaluate our SubST systems.

(Graves et al., 2006) module to avoid the speech encoder pre-training (Gaido et al., 2021a). The encoder and decoder embeddings are shared. We select the hyper-parameters of the Fairseq implementation,[22] except for a higher learning rate of $1 \cdot 10^{-3}$ since pre-training was skipped. The vocabulary is generated using SentencePiece (Kudo and Richardson, 2018), setting the size to 10k unigrams both for the mono- and multilingual segmenters.

For the *Supervised* baseline using OpenSubtitles data, we follow the data selection process for the highest-performing segmenter in (Karakanta et al., 2020c) (*OpenSubs-42*). We first filter sentences with subtitles of a maximum of 42 characters. Since line breaks are not present in OpenSubtitles, we substitute `<eob>` symbols with `<eol>` with a probability of 0.25, paying attention not to insert two consecutive `<eol>`. This proportion reflects the `<eol>`/`<eob>` distribution featured by the MuST-Cinema training set. We noted that almost 90% of the sentences filtered contain only one subtitle. This is not very informative for the segmenter, since the only operation required is inserting one `<eob>` at the end of the sentence. For this reason, we further select only sentences with at least two subtitles (or two subtitle lines). This results in 2,956,207 sentences for es and 683,382 sentences for nl. We then add the same number of sentences containing only one subtitle. After this process, we obtain 5,912,414 sentences for es and 1,366,764 sentences for nl. The supervised baseline is trained with the same settings as the textual monolingual segmenter.

For the *Count Chars* baseline, a break is inserted before reaching the 42-character limit, as per TED guidelines. If the 42-character limit is reached in the middle of a word, the break is inserted before this word. This method will always obtain a 100% conformity to the length constraint. As with the data filtering process, `<eol>` is inserted with a probability of 0.25.

For the SubST models, we use the speech-to-text task *small* architecture of Fairseq with the additional CTC module as in (Papi et al., 2021a).

We use 4 GPUs K80 for training all the architectures: it takes around 1 day for the textual-only and around 1 week for the multimodal segmenters and the SubST models. All results are obtained by averaging 7 checkpoints (best, three preceding and three succeeding checkpoints).

Evaluation. To evaluate both the quality of the SubST output and the accuracy of our segmenters, we resort to reference-based evaluation. For translation quality of the SubST output, we use sacreBLEU (Post, 2018)[23], computed on the text from which the

[22] `https://github.com/pytorch/fairseq/blob/main/examples/speech_to_text/docs/mustc_example.md`

[23] BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.1

subtitle breaks are removed. For segmentation accuracy, we use *Sigma* (Karakanta et al., 2022), a novel subtitle segmentation metric based on BLEU. Sigma is the ratio of the segmentation achieved for a given text to the best segmentation that could be achieved. Contrary to other standard segmentation metrics, such as F1, it can be computed when the output text is different from the reference text. To ensure that the system does not over- or under-generate subtitle breaks, we additionally report *Break coverage* computed as follows:

$$Coverage(\%) = \left(\frac{\#\texttt{<break>}_{pred}}{\#\texttt{<break>}_{ref}} \cdot 100 \right) - 100$$

where `<break>` corresponds to either `<eol>` or `<eob>`. EOL and EOB coverage obtains negative values when the segmenter inserts fewer breaks than required or positive values when it inserts more. Lastly, we use *length conformity* (or characters per line – CPL), corresponding to the percentage of subtitles not exceeding the allowed maximum length of 42 CPL, as per TED guidelines.[24]

RESULTS

4.2.1.4.1 Segmentation on *seen* languages

We train the mono/multi-lingual versions of our *Textual/Multimodal* segmenters for the four languages (de, en, fr, it) and measure their performance in terms of Sigma and CPL. The results are shown in Table 4.1.

Looking at the Sigma values, both the *Textual* and the *Multimodal* segmenter perform better than the rule-based baseline, despite a small drop in CPL. The *Multimodal* segmenter always outperforms the *Textual* one by 2 Sigma points on average and inserts break symbols more accurately. Moreover, it benefits from multilingual training in all languages. In contrast, overall subtitle conformity is higher for the *Textual* segmenter in 3 out of 4 languages, where its CPL scores are 1.2-2.6 percentage points above those obtained by the *Multimodal* one. In addition, except for one case (German), higher CPL values are obtained with monolingual training.

4.2.1.4.2 Zero-shot segmentation

Aiming to build a SubST model for unseen languages (es, and nl), we first select the best segmenter for generating synthetic en→{es, nl} data. As shown in Table 4.2, all the models that receive only text as input (*Count Chars*, *Supervised* and *Textual*)

[24] `https://www.ted.com/participate/translate/subtitling-tips`

Segmenter	Training	English		French		German		Italian	
		Sigma	CPL	Sigma	CPL	Sigma	CPL	Sigma	CPL
Count Chars	-	63.71	100%	62.87	100%	62.34	100%	61.49	100%
Textual	mono	84.87	**96.6%**	83.68	**96.7%**	83.62	90.9%	82.22	90.0%
	multi	85.98	88.5%	84.56	94.3%	84.02	90.9%	83.04	**91.2%**
Multimodal	mono	85.76	94.8%	84.25	93.9%	84.22	**91.4%**	82.62	89.9%
	multi	**87.44**	95.0%	**86.49**	94.1%	**86.4**	89.9%	**85.33**	90.0%

Table 4.1: Segmentation results on *seen* languages.

achieve low segmentation performance, with Sigma ranging between 63-75. The zero-shot *Textual* segmenter achieves higher segmentation quality compared to the *Count Chars* and *Supervised* baselines by 10 points. However, its main drawback is the inability to copy the actual text, as shown by the BLEU values of 61 for nl and 69 for es. In this respect, the baselines perform much better. Despite being trained on subtitling data for the particular language, the low segmentation performance of *Supervised* can be attributed to the different domain compared to the MuST-Cinema test set. For example, MuST-Cinema mainly contains long sentences with multiple breaks, while in OpenSubtitles we rarely come across sentences with more than three breaks. Moreover, both *Supervised* and *Textual* generate subtitles conforming to the CPL constraint in only 70% of the cases, despite having received only length-conforming subtitles as training data. The negative values of EOL and EOB coverage show that all textual methods under-generate subtitle breaks. From these results, we can conclude that zero-shot segmentation does not perform satisfactorily with textual input only.

Dutch					
Segmenter	BLEU	Sigma	CPL	EOL	EOB
Count Chars	**100**	63.2	**100%**	**-21.2%**	-7.1%
Supervised	89.5	64.4	71.2%	-31.4%	-51.3%
Textual	61.3	74.4	77.8%	-23.4%	-9.9%
Multimodal	99.9	**80.3**	91.4%	-27.2%	**+0.4%**
Spanish					
Segmenter	BLEU	Sigma	CPL	EOL	EOB
Count Chars	**100**	63.2	**100%**	-24.6%	**-4.4%**
Supervised	92.6	64.1	71.2%	-32.3%	-45.4%
Textual	69.6	75.8	70.1%	-47.6%	-19.3%
Multimodal	99.6	**78.7**	91.8%	**-22.4%**	+4.7%

Table 4.2: Segmentation results on *unseen* languages.

In comparison, the *Multimodal* segmenter performs significantly better. It reaches an

absolute gain of 6.1 Sigma points for nl and 2.9 for es compared to *Textual*. Moreover, contrary to *Textual* and *Supervised*, the *Multimodal* model learned to perfectly copy the text, as shown by the high BLEU scores (up to 99.9 on nl), close to the maximum score of a method – *Count Chars* – that by design does not change the original text. The CPL results are in agreement with BLEU: for both languages, the *Multimodal* model respects the length constraint in more than 91% of the subtitles. Strikingly, even if the two target languages were never seen by the model, these results are similar to those obtained on seen languages (see Table 4.1). Unlike the rest of the models, *Multimodal* is the only model which does not under-generate `<eob>`. This is in line with the results of (Karakanta et al., 2020a), who showed that exploiting the audio in ST is beneficial for inserting subtitle breaks (`<eob>`, for instance, typically corresponds to longer speech pauses). The results are more discordant for the EOL Coverage. On es, *Multimodal* shows a lower tendency to under-generate, while on nl both models fail to insert at least the 23.4% of `<eol>`. We assume this phenomenon is caused by the lower frequency of `<eol>` in the corpus since a subtitle can be composed of only one line, as well as by the higher difficulty in placing the break for which the system cannot resort to speech clues (e.g. pauses).

Ablation. To test the effectiveness of the *Multimodal* model also in the absence of similar languages in the training set, we train it on a limited set of Latin languages (Italian and French) and test it on Dutch, which is a Germanic language.

The results (*fr, it only*) are shown in Table 4.3. Even if trained on only two languages from a different language group, the *fr, it only Multimodal* model shows competitive results. In terms of segmentation, there is only a slight degradation of 3 Sigma points compared to the full multilingual *Multimodal* model and a 3.6% drop in CPL conformity, which could be attributed to a lower EOL coverage. However, it is still significantly better in terms of Sigma, CPL conformity and EOB coverage compared to all the other segmenters (*Count Chars*, *Supervised*, and *Textual*). In terms of changes to the text, as shown by BLEU, it is on par with *Supervised*, a model trained only on Dutch subtitles, and better than the *Textual* by 25 BLEU points. The presence of related languages seems to help the model better copy the text since the main drop compared to the full *Multimodal* model is in terms of BLEU. Overall, we can conclude that the presence of related languages in the training set can enhance the performance, but the segmentation accuracy and conformity are only minimally affected. The results obtained by the *fr, it only Multimodal* confirm the ability and superiority of this model in segmenting texts on unseen languages also belonging to different language groups.

Segmenter	BLEU	Sigma	CPL	EOL	EOB
Count Chars	**100**	63.2	**100%**	**-21.2%**	-7.1%
Supervised	89.5	64.4	71.2%	-31.4%	-51.3%
Textual	61.3	74.4	77.8%	-23.4%	-9.9%
Multimodal	99.9	**80.3**	91.4%	-27.2%	**+0.4%**
- fr, it only	*88.9*	*77.0*	*87.8%*	*-34.8%*	***-0.4%***

Table 4.3: Ablation results on MuST-Cinema amara en→nl. All but the last line are from Table 4.2.

4.2.1.4.3 SubST with Synthetic Data

Since our *Multimodal* segmenter achieves the best performance overall, we use it to automatically generate the synthetic counterpart of the en→{es, nl} sections of MuST-Cinema. The resulting data is respectively used to train two SubST systems. The goal is to achieve comparable performance to that of similar models trained on manually segmented subtitles. For this purpose, using the same architecture, we also train two systems on the original manual segmentations of MuST-Cinema.

Dutch					
Data	BLEU	Sigma	CPL	EOL	EOB
Original	**25.3***	**81.58**	91.2%	-36.8%	+8.0%
Synthetic	24.3*	75.52	**94.7%**	**-20.4%**	**+4.8%**
Spanish					
Data	BLEU	Sigma	CPL	EOL	EOB
Original	**30.7***	**79.21**	**96.7%**	**-10.0%**	+10.9%
Synthetic	**30.7***	77.84	94.2%	-21.5%	**+9.9%**

Table 4.4: Results of the SubST systems. The * stands for statistically **not** significant results according to the bootstrap resampling test (Koehn, 2004)

As shown in Table 4.4, the SubST system trained on our automatically segmented data (*Synthetic*) shows comparable performance with the system trained on the original segmentation (*Original*). The BLEU between the two models is identical for es, while for nl the difference is not significant. On the contrary, the Sigma for the system trained on manual segmentations is higher than for the synthetic ones by 6 points for nl but less than 2 for es. These results highlight that the breaks introduced by a non-perfect automatic segmentation influence the way the subtitle breaks are placed in the translation but not necessarily the translation itself. For the length constraint, both systems obtain high CPL conformity, with the *Synthetic* model scoring 3.5% more on nl

and 2.5% less on es. This is related to the number of <eol> and <eob> inserted by the system: the more subtitle breaks are present, the more fine-grained the segmentation, leading to higher conformity. Indeed, CPL is higher when the Break Coverage is high.

Manual Analysis. Upon examination of the segmentation patterns of the two en→es systems,[25] we did not identify particular differences. Specifically, the inserted <eob> tags follow punctuation marks in 76% of the cases for both models and are followed by prepositions and conjunctions in 32% and 29% for *Original* and *Synthetic* respectively. Similar patterns between outputs were observed for <eol> too, which is followed by a comma in the majority of cases and by the same function words as <eob>. These results suggest that systems trained on automatically segmented data are able to reproduce similar segmentation patterns to those trained on original data without showing a significant degradation in the translation.

CONCLUSIONS

We presented an automatic segmenter able to turn existing ST corpora into SubST training data. Through comparative experiments on two language pairs in zero-shot conditions, we showed that SubST systems trained on this synthetic data are competitive with those built on human-annotated subtitling corpora. Building on these positive results, and conditioned to the availability of suitable benchmarks, verifying the portability of the approach to a larger set of languages and domains is our priority for future work.

[25] We were unable to replicate the analysis on nl as we do not have the required linguistic competencies.

with Automatically Segmented ST Corpora

4.2.2 PAPER #2

Direct Speech Translation for Automatic Subtitling

INTRODUCTION

With the growth of websites and streaming platforms such as YouTube and Netflix,[26] the amount of audiovisual content available online has dramatically increased. Suffice to say that the number of hours of Netflix original content has increased by 2,400% from 2014 to 2019.[27] This phenomenon has led to a huge demand for subtitles, which is becoming more and more difficult to satisfy only with human resources. Consequently, automatic subtitling tools are spreading to reduce subtitlers' workload by providing them with suggested subtitles to be post-edited (Álvarez et al., 2015; Vitikainen and Koponen, 2021). In general, subtitles can be either *intralingual* (hereinafter *captions*), if source audio and subtitle text are in the same language, or *interlingual* (hereinafter *subtitles*), if the text is in a different language. In this paper, we focus on automatizing interlingual subtitling, framing it as a speech translation (ST) for the subtitling problem.

Differently from ST, in automatic subtitling, the generated text has to comply with multiple requirements related to its length, format, and the time it should be displayed on the screen (Cintas and Remael, 2021). These requirements, which depend on the type of video content and target language, are dictated by the need to keep users' cognitive effort as low as possible while maximizing comprehension and engagement (Perego, 2008; Szarkowska and Gerber-Morón, 2018). This often leads to a condensation of the original spoken content, aimed at reducing the time required for reading subtitles while increasing that of watching the video (Burnham et al., 2008; Szarkowska et al., 2016).

Being such a complex task, automatic subtitling has so far been addressed by dividing the process into different steps (Piperidis et al., 2004; Melero et al., 2006; Matusov et al., 2019; Koponen et al., 2020; Bojar et al., 2021): automatic speech recognition (ASR), timestamp extraction from audio, segmentation into captions, and their machine translation (MT) into the final subtitles. More recently, drawing from the evidence that direct models achieve competitive quality with cascade architectures (Ansari et al., 2020), Karakanta et al. (2020a) proposed an ST system that jointly translates and segments into subtitles, arguing that direct models are able to better exploit speech cues and prosody

[26] https://www.insiderintelligence.com/insights/ott-video-streaming-services/
[27] https://www.statista.com/statistics/882490/netflix-original-content-hours/

in subtitle segmentation. However, their system does not generate timestamps, hence missing a critical aspect to reach the goal of fully automatic subtitling. Furthermore, the current lack of benchmarks hinders a thorough evaluation of the technologies developed for automatic subtitling. In fact, the only corpus publicly available to date is MuST-Cinema (Karakanta et al., 2020b), which contains only single-speaker audios in the TED-talks domain with verbatim translations.

To fill these gaps, this paper presents the first automatic subtitling system that performs the whole task with a single direct ST model and introduces two new benchmarks. Our contributions can be summarized as follows:

- We propose the first direct ST model for automatic subtitling able to produce both subtitles and timestamps. Code and pre-trained models are released under the Apache License 2.0 at: `https://github.com/hlt-mt/FBK-fairseq/`;
- We introduce two (en→{de, es}) benchmarks for automatic subtitling, covering new domains, news/documentaries and interviews, with the presence of background noise and multiple speakers. We release them under the CC BY-NC 4.0 license at: `https://mt.fbk.eu/ec-short-clips/` and `https://mt.fbk.eu/europarl-interviews/`;
- We conduct the first extensive comparison between automatic subtitling systems based on cascade and direct ST models on all the 7 language pairs of MuST-Cinema (en→{de, es, fr, it, nl, pt, ro}), showing the superiority of our direct solution, while also demonstrating its competitiveness with production systems on both MuST-Cinema and out-of-domain benchmarks.

BACKGROUND

4.2.2.2.1 Direct Speech Translation

While the first cascaded approach to ST was proposed decades ago (Stentiford and Steer, 1988; Waibel et al., 1991), direct models[28] have recently become increasingly popular (Bérard et al., 2016; Weiss et al., 2017) due to their ability to avoid error propagation (Sperber and Paulik, 2020), their superior exploitation of prosody and better audio comprehension (Bentivogli et al., 2021), and their lower computational cost (Weller

[28]According to the official IWSLT definition (`https://iwslt.org/2023/offline`), a direct model is a system that does not use intermediate discrete representations to generate the outputs from audio segments and whose parameters used during decoding are all trained altogether on the ST task, while it does not consider the audio segmentation.

et al., 2021). Motivated by these advantages, direct models are rapidly evolving and their initial performance gap with cascade architectures (Niehues et al., 2019) has been significantly reduced, leading to a substantial parity in the latest IWSLT campaigns (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022). Such improvements can be partly attributed to the development of specialized architectures for speech processing (Chang et al., 2020; Papi et al., 2021b; Burchi and Vielzeuf, 2021; Kim et al., 2022; Andrusenko et al., 2022), which are all variants of a Transformer model (Vaswani et al., 2017) preceded by convolutional layers that reduce the length of the input sequence (Bérard et al., 2018; Di Gangi et al., 2019c). Among them, Conformer (Gulati et al., 2020) is currently the best-performing model in ST (Inaguma et al., 2021a). For this reason, we build our systems with this architecture and test, for the first time, its effectiveness in the challenging task of fully automatic subtitling.

4.2.2.2.2 Subtitling Requirements

Subtitles are short pieces of timed text, generally displayed at the bottom of the screen, which describe, transcribe, or translate the dialogue or narrative. A subtitle is composed of two elements: the text, shown into "blocks", and the corresponding start and end display time – or timestamps.[29]

Depending on the subtitle provider and the audiovisual content, different requirements have to be respected concerning both the text space and its timing. These constraints typically consist in: *i)* using at most two lines per block; *ii)* keeping linguistic units (e.g. noun and verb phrases) in the same line; *iii)* not exceeding a pre-defined number of characters per line (CPL), spaces included; *iv)* not exceeding a pre-defined reading speed, measured in number of characters per second (CPS). While a typical value used as the maximum CPL threshold is 42 for most Latin languages,[30] there is no agreement on the maximum CPS allowed. For instance, Netflix guidelines[31] allow up to 17 CPS for adults and 15 for children programs, TED guidelines[32] up to 21 CPS, and Amara guidelines[33] up to 25 CPS.

To convey the meaning of the audiovisual product while adhering to time and space constraints, in some domains and scenarios, subtitles require compression or condensation

[29]The most widespread subtitle format is SubRip or srt.

[30]`https://www.ted.com/participate/translate/subtitling-tips`

[31]`https://partnerhelp.netflixstudios.com/hc/en-us/articles/219375728-Timed-Text-Style-Guide-Subtitle-Templates`

[32]`https://www.ted.com/participate/translate/subtitling-tips`

[33]`https://blog.amara.org/2020/10/22/create-quality-subtitles-in-a-few-simple-steps/`

(Kruger, 2001; Gottlieb, 2004; Aziz et al., 2012; Liu et al., 2020a; Buet and Yvon, 2021). Due to the rehearsed nature of TED talks, the subtitles in MuST-Cinema have a limited degree of condensation, and the translation is mostly verbatim. In addition, the audio conditions (no background noise and a single speaker) are not representative of all the diverse contexts where subtitling is applied, such as news and movies. To fill this gap, we introduce two new benchmarks that feature different domains, scenarios (e.g., multiple speakers), and levels of subtitle condensation.

4.2.2.2.3 Automatic Subtitling

Attempts to (semi-)automatize the subtitling process have been done with cascade systems made of an ASR, a segmenter, and an MT model. Most works focused on adapting the MT module to subtitling with the goal of producing shorter and compressed texts. This has been performed either using statistical approaches trained on subtitling corpora (Volk et al., 2010; Etchegoyhen et al., 2014; Bywood et al., 2013) or by developing specifically tailored decoding solutions on statistical (Aziz et al., 2012) and neural models (Matusov et al., 2019). In particular, recent research efforts focused on controlling the MT output length so as to satisfy isometric requirements between source transcripts and target translations (Lakew et al., 2019; Matusov et al., 2020; Lakew et al., 2021, 2022). In addition, (Öktem et al., 2019; Federico et al., 2020; Virkar et al., 2021; Tam et al., 2022; Effendi et al., 2022) proved the usefulness of injecting prosody information about speech cues, such as pauses, in determining subtitle boundaries. Given the possibility for direct ST systems to access this information and their advantages mentioned in Section 4.2.2.2.1, Karakanta et al. (2020a, 2021a) built the only (to the best of our knowledge) automatic subtitling system using a direct ST model, confirming with their results that the ability of direct ST systems to leverage prosody has particular importance for subtitle segmentation. However, their solution only covers the translation and segmentation into subtitles, neglecting the timestamp generation. Our study is hence the first to complete the entire subtitling process with a direct ST model and to evaluate its performance on all aspects of the subtitling task.

METHOD

Motivated by all the advantages discussed in Section 4.2.2.2.1 and 4.2.2.2.3, we build the first automatic subtitling system solely based on a direct ST model (Figure 4.5). Our system works as follows: *i)* the audio is fed to a *Subtitle Generator* (Section 4.2.2.3.1) that produces the (untimed) subtitle blocks; *ii)* the computed encoder representations

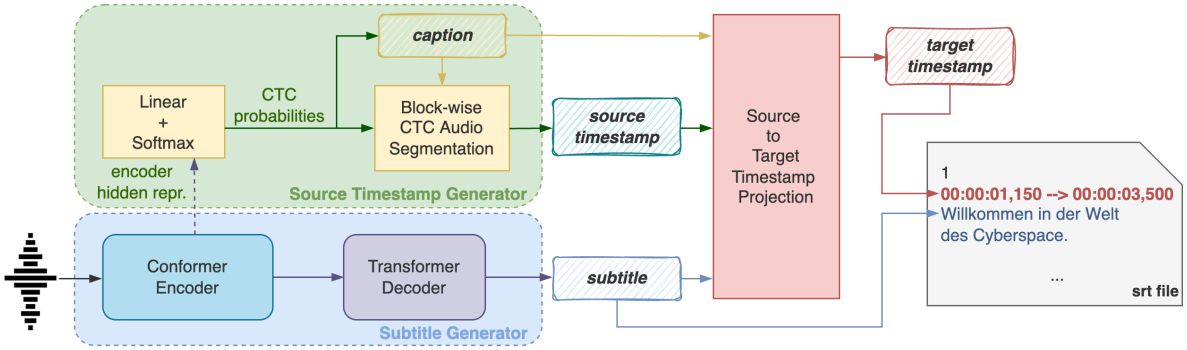


Figure 4.5: Architecture of the direct ST system for automatic subtitling.

are passed to the *Source Timestamp Generator* (Section 4.2.2.3.2) to obtain the caption blocks and their corresponding timestamps; *iii)* the subtitle timestamps are estimated by the *Source-to-Target Timestamp Projection* (Section 4.2.2.3.3) from the generated subtitles, captions, and source timestamps. These modules are described in the rest of this section.

4.2.2.3.1 Subtitle Generation

We train a direct ST Conformer-based model that jointly performs the ST task and the segmentation of the generated translation into (untimed) subtitle blocks and lines. To this end, we add two special tokens to the vocabulary of our system, `<eob>` and `<eol>`, which respectively represent the end of a subtitle block and the end of a line within a block. Both at training and inference time, `<eob>` and `<eol>` are treated as any other token, without giving them different weights or adding specific loss. Additionally, we do not incorporate losses aimed at minimizing the number of generated characters or explicitly optimizing for CPL and CPS compliance.

4.2.2.3.2 Source Timestamp Generation

Estimating timestamps for the generated subtitle blocks from source audio is a challenging task. Current sequence-to-sequence models, in fact, generate target sequences that are decoupled from the input and, therefore, their tokens do not have a clear relationship with the frames they correspond to. To recover this relationship, we start from the observation that direct ST models are often trained with an auxiliary Connectionist Temporal Classification or CTC loss (Graves et al., 2006) in the encoder to improve model convergence (Kim et al., 2017; Bahar et al., 2019). The CTC maps the input frames to the transcripts – in our use case, captions – and we propose to leverage this

CTC module at inference time to estimate the block timestamps.

In particular, the encoder representations computed during the forward pass are fed to the CTC module that provides the frame-level probability distribution over the source vocabulary tokens (including `<eob>`, `<eol>`, and the additional CTC *blank* token). This sequence of CTC probabilities over the source vocabulary serves two purposes. First, it is used to predict the caption with the CTC beam search algorithm (Graves and Jaitly, 2014).[34] Second, it is fed, together with the generated caption, to the CTC-based segmentation algorithm (Kürzinger et al., 2020), whose task is to find the most likely alignment between caption tokens and audio frames. The algorithm builds a trellis over the time steps for the generated tokens and, at each time step, only three paths are possible: *i)* staying at the same token (self-loop); *ii)* moving to the *blank* token; *iii)* moving to the next token. To avoid forcing the caption to start at the beginning of the audio, the transition cost for staying at the first token is set to 0. Otherwise, the transition cost is the CTC-predicted probability for a given token in that time step. The trellis is then backtracked from the time step with the highest probability in the last token of the generated caption until the first token is reached. In our case, since we are interested in the timestamps of the subtitle blocks, we extract block-wise alignments that correspond to the start and the end time of each block. This means finding the time in which the first word of each subtitle is pronounced and the time in which the corresponding `<eob>` symbol is emitted by using the aforementioned algorithm.

4.2.2.3.3 Source-to-Target Timestamp Projection

After generating the untimed subtitles (Section 4.2.2.3.1), and captions with their timestamps (Section 4.2.2.3.2), the next step is to obtain the timestamps for subtitle blocks on the target side. In general, caption and subtitle segmentation may differ for many reasons (e.g. due to different syntactic patterns between languages) and imposing the caption segmentation on the subtitle side – as done in most cascade approaches (Georgakopoulou, 2019; Koponen et al., 2020) – could be a sub-optimal solution. For this reason, we introduce a caption-subtitle alignment module that projects the source timestamps to the target blocks. To perform this task, we tested the three alternative methods described below.

Block-Wise Projection (BWP). This method operates at character level to project the predicted source-side (captions) timestamps on the target side (subtitles) without

[34]We also tested greedy decoding, in which the most likely label for each time step is chosen to obtain the output sequence. However, this approach did not prove effective.

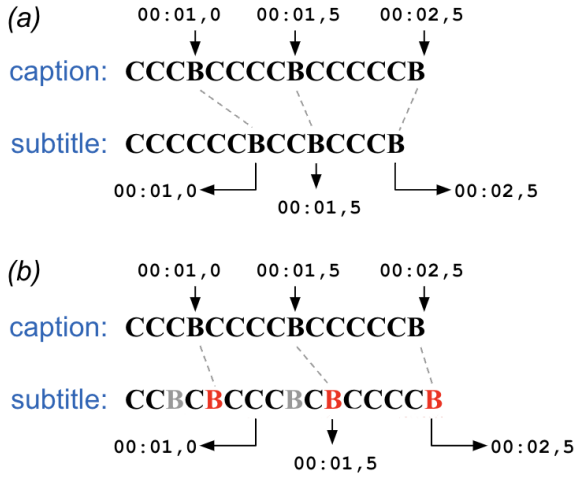


Figure 4.6: Example of BWP projection with *(a)* same number of blocks and *(b)* different number of blocks between caption and subtitle.

alterations. When the number of caption and subtitle blocks is equal, a condition that occurs in ∼80% of the cases, the timestamps of each caption block are directly assigned to the corresponding subtitle block.[35] This process is depicted in Figure 4.6.a, in which "C" and "B" respectively stand for characters and blocks in the caption and subtitle. When the number of caption and subtitle blocks is different (Figure 4.6.b), the target segmentation is discarded and replaced with the caption segmentation. In this case, line and block boundaries (<eol>/<eob>) are inserted in the target side by matching the number of characters each line/block has in the caption. If the insertion falls in the middle of a word, the <eol>/<eob> is appended to the word. This approach has two main weaknesses. First, it assumes that, when captions and subtitles have the same number of blocks, these blocks contain the same linguistic content, while this is not guaranteed. Second, it ignores the subtitle segmentation in ∼20% of the cases.

Levenshtein-based Projection (LEV). To overcome the above limitations, our second method exploits the Levenshtein distance-based alignment (Levenshtein, 1966) between captions and subtitles. This method estimates the target-side timestamps from the source-side timestamps without ever altering the original target-side segmentation. First, all the non-block characters are masked with a single symbol ("C"). For instance, "*This is a block* <eob>" is converted into "CCCCCCCCCCCCCCCB", where "B" stands for <eob>. Then, the masked caption and subtitle are aligned with the weighted version of Levenshtein distance, in which the substitution operation is forbidden so as to avoid

[35] Selecting the candidates with the closest number of blocks among the source and target n-best lists had negligible effects.

the replacement of a character with a block and vice versa. If the positions of a block in the aligned caption and subtitle match, its caption timestamp is directly assigned to the subtitle block. If they do not match, the timestamps of the subtitle blocks are estimated from the caption timestamps based on the alignment of "B"s and the number of characters. For instance, given the caption "CCCBCCCCBCCCCCB" and the subtitle "CCCCCCBCCBCCCB", the optimal source-target alignment with the corresponding timestamp calculation is shown in Figure 4.7. In detail, the first subtitle block (CCC-CCC-B) is matched with the first two caption blocks (CCCBCCCCB) and the corresponding timestamp (`00:01,5`) is directly mapped. This also happens with the timestamp `00:02,5` of the last caption (BCC-CCCB) and subtitle block (CCCB). For the second subtitle block (CCB), the timestamp (`00:01,9`) is estimated proportionally from the caption (BCC-CCCB) using the character ratio between the orange block and the orange + green blocks.

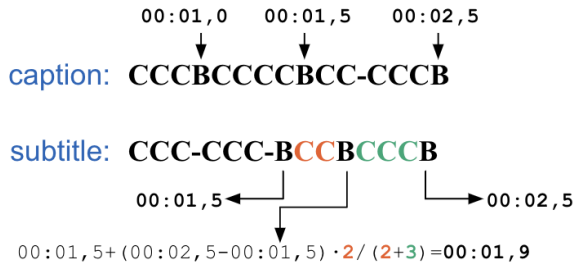


Figure 4.7: Example of Levenshtein-based projection.

Semantic-based Projection (SEM) The third method projects the predicted source-side timestamps on target blocks by looking at the semantic content of the generated captions and subtitles. The method is based on SimAlign (Jalili Sabet et al., 2020), which combines semantic embeddings from fastText (Bojanowski et al., 2017), VecMap (Artetxe et al., 2018), mBERT,[36] and XLM-RoBERTa (Conneau et al., 2020) to align source and target texts at the word level. Specifically, we first align captions and subtitles word by word (`<eol>`/`<eob>` included) with SimAlign. Then, when all `<eob>`s of a subtitle are aligned with `<eob>`s in the caption (66% of the cases), we assign the corresponding timestamp (Figure 4.8). Otherwise, i.e. when at least one `<eob>` in the subtitle is aligned with a caption word or `<eol>` or is not aligned at all, one of the two previous methods is applied as a fallback solution.

[36] `https://github.com/google-research/bert/blob/master/multilingual.md`

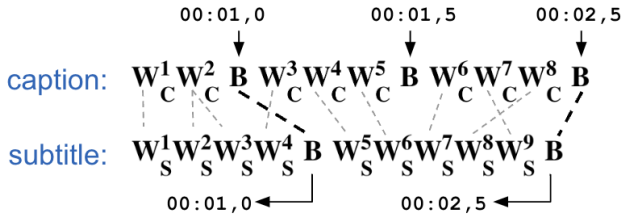


Figure 4.8: Example of Semantic-based projection.

EXPERIMENTAL SETTINGS

4.2.2.4.1 Training Data

For the comparison between cascade and direct architectures (Section 4.2.2.5.2), we train the models in a controlled and easily reproducible data setting by using MuST-Cinema v1.1, the only publicly available subtitling corpus also containing the source speech. It covers one general domain (TED talks), and 7 language pairs, namely en→{de, es, fr, it, nl, pt, ro}. The number of hours in the training set of each language pair is shown in the first row of Table 4.5.

For the comparison with production tools (Section 4.2.2.5.3), we experiment in a more realistic unconstrained data scenario and we focus on en→de and en→es.[37] For training, we use MuST-Cinema, two ST datasets – Europarl-ST (Iranzo-Sánchez et al., 2020) and CoVoST2 (Wang et al., 2020b) – and three ASR datasets – CommonVoice (Ardila et al., 2020), TEDlium Hernandez et al. (2018) and VoxPopuli (Wang et al., 2021). We translate the ASR corpora with the Helsinki-NLP MT models (Tiedemann and Thottingal, 2020) and filter out data with a very high or low transcript/translation character ratio, as per (Gaido et al., 2022e). The use of automatic translations as targets, also known as sequence-level knowledge distillation (Kim and Rush, 2016), is a popular data augmentation method used in the most recent IWSLT evaluation campaigns (Anastasopoulos et al., 2021, 2022) to enhance the performance of ST systems. Since none of the training sets, except for MuST-Cinema, includes the subtitle boundaries (`<eob>` and `<eol>`) in the target translation, we automatically insert them by employing the publicly-released multimodal and multilingual segmenter by Papi et al. (2022c). The segmenter takes the source audio and the unsegmented text as input and outputs the segmented text i.e., containing `<eob>` and `<eol>`. By doing this, we can train our system to jointly translate from speech and segment into subtitles without the need

[37] We select these two language pairs due to, respectively, a different and similar word ordering with respect to the source.

Dataset	de	es	fr	it	nl	pt	ro
MuST-Cinema	388	479	469	441	421	364	410
Europarl-ST	75	74	-	-	-	-	-
CoVoST2	412	412	-	-	-	-	-
CommonVoice	885	885	-	-	-	-	-
TEDlium	444	444	-	-	-	-	-
VoxPopuli	519	519	-	-	-	-	-

Table 4.5: Number of hours of the training sets.

for manually curated subtitle targets, which are hard to find and costly to create. The number of training hours is reported in Table 4.5.

4.2.2.4.2 Test Data

The models are tested in both in-domain and out-of-domain conditions. For in-domain experiments, we use the MuST-Cinema test set, for which we adopt both the original audio segmentation (for reproducibility and for the sake of comparison with previous and future work) and more realistic automatic segmentation obtained with SHAS (Tsiamas et al., 2022). Notice that this audio segmentation is a completely different task from determining subtitle boundaries. Its only goal is splitting long audio files into smaller chunks (or utterances) that can be processed by ST systems, limiting performance degradation due to information loss caused by sub-optimal splits (e.g., in the middle of a sentence). In general, each resulting utterance contains multiple subtitle blocks. For instance, in the MuST-Cinema training set there are ∼2.5 blocks per utterance, even though utterances are quite short (6.4s on average). When automatic segmentation methods like SHAS are applied, this ratio significantly increases, as audio segments are typically much longer, with many segments lasting between 14 and 20 seconds (Gaido et al., 2021c; Tsiamas et al., 2022).

For out-of-domain evaluations, we introduce the two new (en→{de,es}) test sets described below, which we also segment with SHAS.

EC Short Clips. The first test set is composed of short videos from the Audiovisual Service of the European Commission (EC)[38] recorded between 2016 and 2022. These informative clips have an average duration of 2 minutes and cover various topics discussed in EC debates such as economy, environment, and international rights. This benchmark presents several additional difficulties compared to TED talks since the videos often contain multiple speakers, and background music is sometimes present during the speech.

[38] `https://audiovisual.ec.europa.eu/`

We selected the videos with the highest subtitle conformity (at least 80% of the subtitles conforming to 42 CPL, and 75% conforming to 21 CPS), and removed subtitles describing on-screen text. This resulted in 27 videos having a total duration of 1 hour. The target srt files contain ∼5,000 words per language.

EuroParl Interviews. The second test set is compiled from publicly available video interviews from the European Parliament TV[39] (2009-2015). We selected 12 videos of 1 hour total duration, amounting to ∼6,500 words per target language. The videos present multiple speakers and sometimes contain short interposed clips with news or narratives. Apart from the more challenging source audio properties compared to the clean single-speaker TED talks, here the target subtitles are not verbatim and demonstrate a high degree of compression and reduction. As a consequence, the CPL and CPS conformity is very high (∼100%) but this comes at the cost of being more difficult for automatic systems to perfectly match the non-verbatim translations. Nonetheless, to achieve real progress in automatic subtitling, it is particularly relevant to evaluate automatic systems on realistic and challenging benchmarks like the ones we provide.

4.2.2.4.3 Training Settings

Our systems are implemented on Fairseq-ST (Wang et al., 2020a), following the default settings unless stated otherwise. The input is represented by 80 audio features extracted every $10ms$ with sample window of 25 and pre-processed by two 1D convolutional layers with stride 2 to reduce the input length by a factor of 4. All segments longer than $30s$ in the training set are filtered out to speed up training. The models are based on encoder-decoder architectures and composed by a stack of 12 Conformer encoder layers and 8 Transformer decoder layers. We apply CTC loss to the 8^{th} encoder layer and use its predictions to compress the input sequences to reduce RAM consumption (Liu et al., 2020c; Gaido et al., 2021a). Both the Conformer and Transformer layers have a 512 embedding dimension and 2,048 hidden units in the linear layer. We set dropout to 0.1 in the linear, attention, and convolutional modules. In the convolutional modules, we also set a kernel size of 31 for the point- and depth-wise convolutions.

For the comparison between cascade and direct architectures, we train a one-to-many multilingual ST model that prepends a token representing the selected target language for decoding (Inaguma et al., 2019) on all the 7 languages of MuST-Cinema. Conversely, for the comparison with production tools, we develop a dedicated ST model for each target language (de, es). For inference, we set the beam size to 5 for both subtitles and

[39] `https://www.europarltv.europa.eu/`

captions.

We train with Adam optimizer (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$) for 100,000 steps. The learning rate increases linearly up to 0.002 for the first 25,000 warm-up steps and then decays with an inverse square root policy, apart from fine-tunings, where it is fixed at 0.001. Utterance-level Cepstral Mean and Variance Normalization (CMVN) and SpecAugment Park et al. (2019) are applied during training, as per Fairseq-ST default settings. The vocabularies are based on SentencePiece models (Sennrich et al., 2016) with size 8,000 for the source language. For the multilingual model trained on MuST-Cinema, a shared vocabulary is built with a size of 16,000 while, for the two models developed to compare with production tools, we build German and Spanish vocabularies with a size of 16,000 subwords each. The ASR of our cascade model is trained using the same source language vocabulary of size 8,000 used in the translation setting. The MT model is trained using the standard hyper-parameters of the Fairseq multilingual MT task (Ott et al., 2019), with the same source and target vocabularies of the ST task.

For all models, we stop the training when the validation loss does not improve for 10 epochs and the final models are obtained by averaging 7 checkpoints (the best, 3 preceding and 3 succeeding). Training is performed on 4 NVIDIA A100 (40GB RAM), with 40k max tokens per mini-batch and an update frequency of 2, except for the MT models for which 8 NVIDIA K80 (12GB RAM) are used with 4k max tokens and an update frequency of 1. Table 4.6 lists the total number of parameters of our direct models, showing that it is $\sim$1/3 of the cascade system used as a term of comparison.

4.2.2.4.4 Terms of Comparison

We compare our direct ST system with a cascade pipeline trained under the same data conditions and with production tools.

Cascade. We build an in-domain cascade composed of: an ASR, an audio-forced aligner, a segmenter, and an MT system. The ASR has the same architecture as our ST system (Conformer encoder + Transformer decoder), and it is trained on MuST-Cinema transcripts without `<eob>` and `<eol>`. The audio forced aligner used to estimate the timestamps (Gretter et al., 2021) is based on the Kaldi[40] acoustic model. The subtitle segmenter is the same multimodal segmenter we used to segment the training data for the direct system (Section 4.2.2.4.1). The MT is a multilingual model trained on the MuST-Cinema (*transcript*, *translation*) pairs without `<eob>` and `<eol>`. The pipeline

[40] `https://github.com/kaldi-asr/kaldi`

System	Num. params
Direct	124.6M
Cascade	341.9M
- ASR	116.4M
- Audio forced aligner	9.7M
- Segmenter	40.6M
- Multilingual MT	175.2M

Table 4.6: Number of parameters for the direct (both multilingual and monolingual) and cascade systems.

works as follows. The audio is first transcribed by the ASR and word-level timestamps are estimated with the forced aligner. Then, the transcript is segmented into captions with the segmenter and each block timestamp is obtained by averaging the end time of the word before an `<eob>` and the start time of the word after it. The segmented text is then split into sentences according to the `<eob>` and, finally, these sentences are translated by the MT. The `<eob>`s are automatically re-inserted at the end of each sentence while `<eol>`s are added to the subtitle translation using the same segmenter.

Production Tools. As a term of comparison for the unconstrained data condition, we use production tools for automatic subtitling. These tools take audio or video content as input and return the subtitles in various formats, including srt. We test three online tools,[41] namely: MateSub,[42] Sonix,[43] and Zeemo.[44] We also compare with the AppTek subtitling system,[45] a cascade architecture whose ASR component is equipped with a neural model that predicts the subtitle boundaries before feeding the transcripts to the MT component (Matusov et al., 2019). For this system, two variants of the MT model are evaluated: a standard model and a model specifically trained to obtain shorter translations in order to better conform to length requirements (Matusov et al., 2020). Since we are not interested in comparing the tools with each other, all system scores are anonymized.

[41] All outputs were collected in August 2022.
[42] `https://matesub.com/`
[43] `https://sonix.ai`
[44] `https://zeemo.ai/`
[45] `https://www.apptek.com/`

4.2.2.4.5 Evaluation

Translation quality, timing, and segmentation of subtitles are measured with multiple metrics. First, we compute SubER (Wilken et al., 2022),[46] a tailored TER-based metric (the lower, the better) that scores the overall subtitle quality by considering translation, segmentation and timing altogether. We adopt the cased and with punctuation version of the metric since these aspects are crucial for the quality and comprehension of the subtitles. Next, specifically for translation quality, we use SacreBLEU (Post, 2018),[47] on texts from which `<eol>` and `<eob>` have been removed. The quality of segmentation into subtitles is evaluated with Sigma from the EvalSub toolkit (Karakanta et al., 2022). Since BLEU and Sigma require the same audio segmentation between reference and predicted subtitles, we re-align the predictions in case of non-perfect alignment with the mWERSegmenter (Matusov et al., 2005). Lastly, to check the spatio-temporal compliance described in Section 4.2.2.2.2, we compute CPL conformity as the percentage of lines not exceeding 42 characters, and CPS conformity as the percentage of subtitle blocks having a maximum reading speed of 21 characters per second.[48] Confidence intervals (CI) are computed with bootstrap resampling (Koehn, 2004).

RESULTS

In this section, we first (Section 4.2.2.5.1) choose the best timestamp projection method among those introduced in Section 4.2.2.3.3. Then (Section 4.2.2.5.2), we compare the cascade and direct approaches trained in the same data conditions. Lastly (Section 4.2.2.5.3), we show that our direct model, even though trained in laboratory settings, is competitive with production tools. In addition, in Appendix 4.2.2.7.1, we analyze the performance of the CTC-segmentation algorithm for timestamp estimation compared to forced aligner tools.

4.2.2.5.1 Timestamp Projection

The quality of source-to-target timestamp projection is crucial to correctly estimate the target-side timestamps and, in turn, to produce good subtitles. To select the best strategy, we compare the methods in Section 4.2.2.3.3 using the constrained model on

[46] Version 0.2.0.

[47] case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

[48] We used version 1.1 of the script adopted for the IWSLT subtitling task (`https://iwslt.org/2023/subtitling`): `https://github.com/hlt-mt/FBK-fairseq/blob/master/examples/speech_to_text/scripts/subtitle_compliance.py`.

	en-de				en-es			
Model	SubER (↓)	Sigma (↑)	CPL (↑)	CPS (↑)	SubER (↓)	Sigma (↑)	CPL (↑)	CPS (↑)
Gold audio segmentation								
Baseline	63.5	65.6	77.7	64.4	52.0	70.4	80.7	68.0
BWP	60.8	75.6	86.1	64.0	48.6	78.5	90.9	66.9
LEV	**58.7**	**78.8**	**88.8**	**65.4**	**46.7**	**81.1**	93.9	**68.4**
SEM	60.7	75.5	88.6	63.7	48.6	78.8	**94.0**	65.5
Automatic audio segmentation								
Baseline	66.9	62.0	78.2	70.5	55.7	66.0	79.9	75.1
BWP	62.8	73.3	86.2	70.3	51.8	75.9	89.6	73.5
LEV	**60.3**	**78.5**	**88.9**	**72.1**	**48.5**	**80.6**	**94.2**	**76.1**
SEM	62.8	75.8	**88.9**	69.7	51.4	78.3	**94.2**	72.9

Table 4.7: Comparison of timestamp projection methods on the MuST-Cinema en→{de, es} test set.

the MuST-Cinema test sets for en→{de, es}. To test the robustness of the various methods when gold-segmented audio is not available, we also report the results using the automatic audio segmentation in addition to that obtained using the gold one.

Results are shown in Table 4.7. BLEU is not reported because the translated text is always the same, regardless of the timestamp projection method. We also report, as a baseline, a method that completely ignores the target segmentation and always maps the caption segmentation onto the subtitle as in BWP when the number of caption and subtitle blocks is different (Section 4.2.2.3.3). For the SEM method, if the source-target alignment is not found by SimAlign, the LEV method is applied instead.[49]

The results highlight the superiority of the LEV method, which outperforms the others on almost all metrics, with similar trends for both language pairs. The gap is more marked in the realistic scenario of automatically segmented audio, likely due to the fact that the audio segments produced by SHAS are longer than the manually annotated ones (8.6s vs 5.5s). As such, each audio segment contains more blocks to align, so the difference between the methods emerges more clearly. The low scores obtained by the baseline confirm that the caption segmentation is not optimal for the target language. Furthermore, SEM yields results that are either comparable to or slightly better than those obtained by BWP, especially in terms of Sigma and CPL, while being always worse than LEV. In addition, SEM exhibits lower CPS conformity even compared to the baseline. Consequently, its performance suggests that semantically-motivated approaches are not the best solution for timestamp projection.

[49]We also applied the baseline and the BWP method as a fallback method for SEM but it led to worse results.

Focusing on the LEV method, we observe that segmentation quality (higher Sigma) and overall subtitle quality (lower SubER) are slightly better when the gold segmentation is used, as expected. Conversely, CPS conformity is higher with automatic audio segmentation. This counter-intuitive result can be explained as follows: audio segmentation not only splits but sometimes also cuts the audio according to speakers' pauses, while manual segmentation delimits speech boundaries more aggressively than automatic one. In our case, manual segmentation results in audio segments that are about 2% shorter than those obtained with the automatic segmentation, thus "forcing" the generated subtitles to appear on screen for a shorter time, which in turn leads to a higher reading speed.

Sys.	en-de	en-es	en-fr	en-it	en-nl	en-pt	en-ro	Avg.
				SubER (↓)				
Casc.	64.2 (64.2±2.4)	50.5 (50.5±2.2)	57.0 (57.0±1.8)	54.2 (54.2±1.7)	52.8 (52.8±1.8)	49.7 (49.7±1.7)	52.7 (52.7±2.0)	54.4
Dir.	**58.7** (58.7±2.3)	**46.7** (46.7±2.1)	**52.9** (52.9±1.7)	**50.4** (50.4±1.7)	**47.4** (47.4±1.9)	**44.6** (44.6±1.7)	**48.5** (48.5±2.1)	**49.9**
				BLEU (↑)				
Casc.	18.9 (18.9±1.4)	32.4 (32.4±1.8)	25.1 (25.1±1.5)	26.0 (26.0±1.6)	25.8 (25.8±1.5)	31.4 (31.4±1.7)	28.4 (28.3±1.6)	26.9
Dir.	**22.1** (22.1±1.6)	**35.9** (35.8±1.9)	**28.0** (28.0±1.6)	**29.6** (29.6±1.8)	**31.6** (31.6±1.8)	**36.8** (36.7±1.7)	**31.9** (31.8±1.8)	**30.8**
				Sigma (↑)				
Casc.	**79.5** (79.5±2.0)	80.9 (80.9±1.5)	84.0 (84.0±1.7)	83.8 (83.8±1.6)	77.5 (77.4±1.8)	81.2 (81.2±1.7)	**86.4** (86.4±1.5)	81.9
Dir.	78.8 (78.8±2.0)	**81.1** (81.1±1.5)	**84.1** (84.1±1.7)	**85.1** (85.1±1.5)	**83.1** (83.1±1.6)	**84.5** (84.4±1.4)	85.3 (85.3±1.4)	**83.1**
				CPL (↑)				
Casc.	81.8 (81.8±1.9)	83.4 (83.3±1.8)	85.2 (85.2±1.7)	81.4 (81.4±1.9)	83.3 (83.2±1.9)	78.1 (78.1±2.0)	53.3 (53.3±3.0)	78.1
Dir.	**88.9** (88.9±1.5)	**94.0** (94.0±1.1)	**91.9** (91.9±1.2)	**89.3** (89.2±1.5)	**84.0** (84.0±1.8)	**88.2** (88.2±1.5)	**92.1** (92.1±1.2)	**89.8**
				CPS (↑)				
Casc.	**69.1** (69.1±2.6)	**74.0** (73.9±2.7)	**64.3** (64.3±2.9)	**71.2** (71.2±2.8)	**74.4** (74.4±2.5)	**74.7** (74.7±2.6)	**76.2** (76.2±2.4)	**72.0**
Dir.	65.4 (65.4±2.7)	68.4 (68.3±2.7)	60.7 (60.8±2.8)	67.9 (67.9±2.6)	72.2 (72.2±2.6)	71.9 (71.8±2.7)	76.0 (75.9±2.4)	68.9

Table 4.8: Cascade (Casc.) and direct (Dir.) results on all MuST-Cinema language pairs with 95% CI in parentheses.

4.2.2.5.2 Cascade vs. Direct

After selecting LEV as our best timestamp projection method, we evaluate cascade and direct ST systems trained in the same data condition. Before this, to ensure the competitiveness of our cascade baseline, we compare it with the results obtained on the MuST-Cinema test set by the other cascade systems presented in literature, namely: en→{de, fr} by Karakanta et al. (2021a), and en→fr by Xu et al. (2022). As these works report only BLEU with breaks, that is BLEU computed including also `<eob>` and `<eol>`, we compare our cascade baseline with them on that metric.[50] Although these works

[50] `<eob>` and `<eol>` are considered as a single token and replaced, respectively, with § and µ as in the EvalSub toolkit.

leverage large additional training corpora for both ASR (e.g. LibriSpeech – (Panayotov et al., 2015)) and MT (e.g. OPUS – (Tiedemann, 2016) – and WMT-14 – (Bojar et al., 2014)), our cascade trained only on MuST-Cinema performs on par with them. It scores 20.2 in German and 26.2 in French, which are similar or even better than, respectively, 19.9 and 26.9 of (Karakanta et al., 2021a), and 25.8 in French of (Xu et al., 2022). These results confirm the strength of our baselines and the soundness of our experimental settings.

Table 4.8 reports the scores of the constrained direct and cascade models. The overall subtitle quality of the direct solution is significantly higher compared to that of the cascade on all language pairs, with a SubER decrease of 3.8-5.5 points, corresponding to an ∼8% improvement on average. Since SubER measures translation, segmentation and timestamp quality altogether, to disentangle the contribution of each of these aspects we leverage the other metrics. The higher Sigma of our system (+1.2 average improvement) demonstrates that the joint generation of subtitle content and boundaries results in superior segmentation. This finding corroborates previous research on the value of prosody (see Section 4.2.2.2.3), and the ineffectiveness of projecting caption segmentation onto subtitles, as done by cascade approaches (Georgakopoulou, 2019; Koponen et al., 2020). The sub-optimal placement of block boundaries in the cascade system can also account for the superior translation quality of our method (+3.9 BLEU average improvement): as the MT component translates the caption block-by-block, inaccurate boundaries can impede access to information required for proper translation.

Looking at the conformity metrics, the direct system complies with the length requirement of 42 characters (CPL) in almost 90% of cases while the cascade system does so in only 78.1%. This difference is explained by the higher number of `<eol>` generated by the direct model (10-15% more than the cascade), although it is still lower than that of the reference (8-10% less). According to the statistics computed on the outputs of the two systems, the cascade does not only have a higher average number of characters per line (32 vs. 29), but its variance is 1.5-2 times greater, with lines sometimes close to or even longer than 100 characters on all language pairs. In contrast, most of the CPL violations of the direct system are caused by lines shorter than 60 characters, and lines never exceed 70 characters. The trend for CPS is instead different since the cascade generates subtitles with a higher conformity to the 21-CPS reading speed (72.0 vs 68.9). This can be partially explained by looking at the generated timestamps: upon a manual inspection of 100 subtitles, we noticed that the direct model tends to assign the start times of the subtitles slightly after those of the cascade (within 100ms of difference), and end times slightly before those of the cascade (mostly within 200ms). Overall, on

the MuST-Cinema test sets, this leads to a total of ∼2,940s with subtitles on the screen for the cascade and ∼2,850s for the direct (∼3% lower).

To sum up, our direct system proves to be the best choice to address the automatic subtitling task in constrained data conditions, reaching better translation quality and more well-formed subtitles. Our results also indicate that improving the reading speed of the generated subtitles is one of the main aspects on which to focus future works.

4.2.2.5.3 Comparison with Production Tools

To test our approach in more realistic conditions, we train our models on several openly available corpora (unconstrained condition) and compare them with production tools, which represent very challenging competitors as they can leverage large proprietary datasets. We focus on two language pairs (en→{de, es}) for both the in-domain MuST-Cinema and in the two out-of-domain EC Short Clips and EuroParl Interviews test sets. We feed all systems with the full test audio clips, so each system has to segment its audio. Only in the case of EC Short Clips, and EuroParl Interviews, we clean the audio using Veed[51] before processing it, for the sake of a fair comparison with production tools that have similar procedures.[52] The impact of audio cleaning is analyzed in Appendix 4.2.2.7.2.

MuST-Cinema. The results of the unconstrained models on the in-domain MuST-Cinema test set are shown in Table 4.9. Compared to production tools, our system shows better translation and segmentation quality as well as a significantly better overall quality on both languages. Gains in BLEU are more evident in Spanish, where we obtain a ∼6% improvement compared to the second-best model (System 4). Also, considerable Sigma improvements are observed with gains of 5.3-34.5% for German and 2.9-24.2% for Spanish, which are in line with SubER improvements of, respectively, 2.6-12.0% and 8.8-27.6%. A perfect CPL conformity is reached by System 1 and 2 for both languages, while our system is on par with System 3 on en-es and falls slightly behind System 3 and 4 on en-de, with a ∼90% average conformity for the two language pairs. System 5 is by far the worst, as it violates the 42 CPL constraint in more than 50% of the lines. As for CPS conformity, we observe that our system achieves better scores compared to System 1 and 5 but it is worse than System 2, 3, and 4 in both language directions, highlighting again the need to improve this aspect in future work.

[51] `https://www.veed.io/`.

[52] E.g., see `https://www.apptek.com/post/asr-in-captions-accessibility-series-article-7` and `https://sonix.ai/articles/how-to-remove-background-audio-noise`.

en-de					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	66.9 (66.9±2.8)	20.1 (20.2±1.5)	71.7 (71.6±2.4)	**100** (100±0.0)	58.7 (58.6±3.1)
System 2	61.5 (61.5±2.4)	22.3 (22.2±1.6)	71.8 (71.8±2.3)	**100** (100±0.0)	76.2 (76.2±2.7)
System 3	68.1 (68.1±1.5)	13.5 (13.5±1.2)	62.1 (62.0±2.6)	91.6 (91.7±1.4)	**89.3** (89.3±1.8)
System 4	67.5 (67.1±7.3)	23.3 (23.2±1.7)	57.9 (57.9±2.2)	96.4 (96.4±0.9)	83.7 (83.7±2.3)
System 5	66.8 (66.8±2.9)	19.5 (19.5±1.5)	74.0 (74.0±2.0)	44.1 (42.8±3.0)	50.2 (50.2±3.1)
Ours	**59.9** (59.9±3.2)	**23.4** (23.4±1.6)	**77.9** (78.0±2.1)	86.9 (86.9±1.6)	68.6 (68.6±2.7)
en-es					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	52.2 (52.2±2.7)	33.4 (33.3±1.8)	76.9 (76.9±1.9)	**100** (100±0.0)	64.6 (64.6±2.9)
System 2	51.3 (51.2±2.4)	32.7 (32.6±1.8)	77.1 (77.0±2.0)	**100** (100±0.0)	77.6 (77.6±2.5)
System 3	58.3 (58.3±1.7)	23.3 (23.2±1.4)	66.1 (66.0±2.3)	94.1 (94.1±1.2)	**87.1** (87.1±2.0)
System 4	53.8 (53.8±4.7)	35.3 (35.3±2.0)	65.7 (65.7±1.8)	81.3 (81.3±2.2)	86.2 (86.1±2.2)
System 5	64.6 (64.6±2.0)	18.6 (18.6±1.3)	79.3 (79.3±1.9)	48.5 (48.5±3.0)	63.0 (62.9±2.8)
Ours	**46.8** (46.7±2.2)	**37.4** (37.5±2.0)	**81.6** (81.7±1.5)	93.2 (93.3±1.1)	74.6 (74.6±2.5)

Table 4.9: Unconstrained results on MuST-Cinema with 95% CI in parentheses.

EC Short Clips. This out-of-domain test set presents additional difficulties compared to TED talks, namely the presence of multiple speakers and background music during speech. It is worth mentioning that our direct ST models have not been trained to be robust to these phenomena, as they are not present in the training data, whereas production tools are designed to deal with any condition, and may have dedicated modules to handle them.

Nevertheless, the results in Table 4.10 show that, even in these challenging conditions, our direct ST models are competitive with production tools on BLEU, Sigma, and SubER. Indeed, there is no clear winner between the systems as the best score for each metric is obtained by a different model, which also varies across languages. Looking at the conformity constraints, Systems 1, 2, and 4 achieve a perfect CPL conformity (100%), while ours is comparable with System 3 and better than System 5. This difference is likely motivated by the number of `<eol>` inserted by our system, which is considerably lower than that of System 4 (368 vs. 635 for German and 451 vs. 594 for Spanish). Instead, the results for CPS conformity follow the same trend observed in the constrained data condition (Section 4.2.2.5.2).

Even though this scenario features completely different domain and audio characteristics, some trends are in line with the results shown in Table 4.9. System 3 always achieves the best CPS conformity, while Systems 1, 2, and 4 achieve perfect CPL conformity on both languages. Moreover, although System 4 achieves the best translation quality

en-de					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	63.0 (63.0±2.4)	23.8 (23.8±1.9)	**71.6** (71.5±2.7)	**100** (100±0.0)	76.1 (76.1±2.8)
System 2	60.8 (60.8±1.8)	22.1 (22.1±1.9)	67.2 (67.1±2.9)	**100** (100±0.0)	91.1 (91.1±1.9)
System 3	**59.0** (58.9±1.9)	25.0 (25.0±1.9)	70.4 (70.4±2.8)	84.6 (84.6±1.9)	**95.4** (95.4±1.4)
System 4	61.5 (61.5±3.3)	**28.2** (28.3±2.0)	59.4 (59.4±2.2)	**100** (100±0.0)	94.9 (95.0±1.5)
System 5	62.4 (62.4±2.2)	24.2 (24.2±1.8)	71.3 (71.2±2.2)	39.8 (39.7±3.4)	71.3 (71.3±3.3)
Ours	59.9 (59.9±2.2)	25.3 (25.3±1.9)	70.8 (70.7±2.4)	81.3 (81.3±2.2)	79.9 (80.0±2.7)
en-es					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	52.9 (52.9±1.8)	33.7 (33.7±1.8)	76.0 (75.9±2.2)	**100** (100±0.0)	80.4 (80.3±2.8)
System 2	51.7 (51.6±1.6)	32.2 (32.3±1.9)	75.6 (75.6±2.2)	**100** (100±0.0)	93.5 (93.5±1.7)
System 3	**49.7** (49.7±1.8)	35.5 (35.5±1.8)	74.9 (74.9±1.9)	87.3 (87.4±1.8)	**95.3** (95.3±1.4)
System 4	50.2 (50.2±2.2)	**39.6** (39.6±1.9)	61.9 (61.9±1.8)	**100** (100±0.0)	93.4 (93.4±1.4)
System 5	64.9 (64.9±1.6)	21.9 (21.9±1.5)	**79.7** (79.6±2.0)	41.7 (41.6±3.3)	73.1 (73.0±3.2)
Ours	52.7 (52.7±2.0)	34.8 (34.9±2.0)	72.6 (72.7±2.0)	88.6 (88.5±1.6)	79.1 (79.0±2.6)

Table 4.10: Unconstrained results on EC Short Clips with 95% CI in parentheses.

(and it is the second best on MuST-Cinema, after our system), its segmentation quality (Sigma) is always the worst, indicating that its subtitles are not segmented in an optimal way to facilitate comprehension. All in all, these results suggest that each production tool has been optimized on a different aspect of automatic subtitling (e.g. System 3 has been optimized to achieve high CPS conformity). In contrast, our direct model, which has been trained without prioritizing any specific aspect, performs on average, also achieving competitive results in out-of-domain scenarios.

EuroParl Interviews. EuroParl Interviews represents the most difficult of the three test sets: it contains multiple speakers, and the target translations are not verbatim since they are compressed to perfectly fit the subtitling constraints (Section 4.2.2.2.2). This characteristic is very challenging for current automatic subtitling tools, especially for our direct model since it has not been trained on similar data.

The results are shown in Table 4.11. As on the EC test set, our system performs competitively with production tools, even achieving the best Sigma for German. For CPL, instead, most systems have high length conformity, even reaching 100%. As already noticed on the other test sets, the CPL conformity is strongly correlated with the number of `<eol>` inserted by a system: our model has an average conformity of 85.5% with only 451 `<eol>` inserted, nearly half of those inserted by System 1 (864), System 2 (711), and System 4 (774) that always comply with the CPL constraint. CPS conformity shows the same trend as with the other test sets.

en-de					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	84.9 (85.0±2.4)	12.3 (12.3±1.1)	64.8 (64.8±2.8)	**100** (100±0.0)	67.6 (67.7±2.8)
System 2	78.4 (78.4±2.0)	13.2 (13.2±1.1)	63.9 (63.9±2.9)	**100** (100±0.0)	79.8 (79.8±2.3)
System 3	**78.1** (78.1±1.9)	13.6 (13.6±1.1)	69.6 (69.6±2.8)	86.9 (86.9±1.6)	**93.2** (93.3±1.4)
System 4	80.1 (80.1±2.7)	**15.8** (15.8±1.3)	56.9 (56.9±2.8)	**100** (100±0.0)	83.8 (83.9±2.2)
System 5	85.1 (85.1±1.9)	11.4 (11.4±1.1)	69.8 (69.8±2.5)	44.4 (44.4±2.8)	59.2 (59.3±2.7)
Ours	80.3 (80.3±2.4)	12.5 (12.5±1.1)	**70.0** (70.0±2.8)	80.9 (81.0±1.9)	68.8 (68.8±2.5)
en-es					
Model	SubER (↓)	BLEU (↑)	Sigma (↑)	CPL (↑)	CPS (↑)
System 1	75.5 (75.5±2.3)	19.8 (19.8±1.3)	72.7 (72.7±2.2)	**100** (100±0.0)	72.7 (72.8±2.5)
System 2	71.4 (71.4±2.1)	20.9 (20.9±1.4)	73.8 (73.8±2.0)	**100** (100±0.0)	81.4 (81.5±2.3)
System 3	70.0 (70.1±2.2)	20.8 (20.8±1.4)	72.8 (72.8±2.0)	90.5 (90.5±1.4)	**93.7** (93.7±1.3)
System 4	**68.6** (68.5±2.5)	**25.4** (25.4±1.4)	61.6 (61.6±2.0)	**100** (100±0.0)	91.5 (91.5±1.8)
System 5	80.8 (80.8±1.7)	13.0 (12.9±1.1)	**77.3** (77.3±2.4)	52.1 (52.1±2.8)	67.4 (67.5±2.7)
Ours	72.3 (72.3±2.2)	20.8 (20.9±1.4)	70.4 (70.4±2.0)	90.1 (90.1±1.3)	76.9 (76.9±2.4)

Table 4.11: Unconstrained results on EuroParl Interviews with 95% CI in parentheses.

Compared to the results in Tables 4.9 and 4.10, we can see that all systems struggle in achieving a comparable overall subtitle quality (SubER), high-quality segmentations (Sigma), and, above all, high translation quality (BLEU). The translation quality of all systems degrades by at least 10 BLEU compared to the values observed on the MuST-Cinema and EC test sets. However, as previously mentioned, these results are expected since the EuroParl Interviews test set contains condensed translations of the source speech.

All in all, we can conclude that our direct ST model, even though not developed as a production-ready system (it is not trained on huge amounts of data and different domains), is competitive with production tools. Indeed, considering the SubER metric computed over the three test sets (Table 4.12), our direct ST approach is the best in both German (67.0) and Spanish (57.2). As only the scores of System 2 fall within the confidence interval of our direct model in both cases, we can conclude that our model is on par with the best production system and outperforms the others in terms of SubER.

	System 1	System 2	System 3	System 4	System 5	Ours
en-de	72.0 (72.0±1.6)	67.2 (67.1±1.3)	69.0 (69.0±1.2)	70.1 (70.1±3.3)	71.9 (71.9±1.7)	**67.0** (67.0±1.7)
en-es	60.3 (60.3±1.5)	58.2 (58.2±1.3)	59.8 (59.8±1.2)	57.8 (57.8±2.4)	70.2 (70.2±1.1)	**57.2** (57.1±1.5)

Table 4.12: SubER (↓) over the three test sets with 95% CI in parentheses.

CONCLUSIONS

In this paper, we proposed the first approach based on direct speech-to-text translation models to fully automatize the subtitling process, including translation, segmentation into subtitles, and timestamp estimation. Experiments in constrained data conditions on 7 language pairs demonstrated the potential of our approach, which outperformed the current cascade architectures with a ∼7% improvement in terms of SubER. In addition, to test the generalisability of our findings across subtitling genres, we extended our evaluation setting by collecting two new test sets for en→{de, es} covering different domains, degrees of subtitle condensation, and audio conditions. Finally, we compared our models with production tools in unconstrained data conditions on both existing benchmarks and the newly collected test sets. This comparison further highlighted that our approach represents a promising direction: although trained on a relatively limited amount of data, our systems achieved comparable quality with production tools, with improvements in SubER ranging from 0.2 to 5.0 on en→de and from 0.6 to 13.0 on en→es over the three test sets.

APPENDIX

4.2.2.7.1 Timestamp Extraction Method

To validate the effectiveness of extracting source-side timestamps with the CTC-based segmentation algorithm, we conduct an ablation study, where we replace it with the forced aligner tool of the Cascade architecture (§4.2.2.4.4). Table 4.13 reports the scores. The forced aligner tool (FA) achieves similar results compared to the CTC-based segmentation algorithm (CTC), with a slightly worse SubER (+0.1) on average on the three test sets. Moreover, it is important to highlight that our method does not require an external model. These findings support our choice and align with previous research by Kürzinger et al. (2020), which highlighted the competitiveness of the CTC-based segmentation approach compared to widely used forced aligners (in their case, Gentle[53]).

4.2.2.7.2 Effect of Background Noise

The presence of background noise in the test sets complicates both the audio segmentation (performed with SHAS) and the generation with the direct ST model. For this reason, for the sake of a fair comparison with production tools, we used Veed to remove the

[53] `https://github.com/lowerquality/gentle`

Method	en-de			en-es			Avg.
	MC	ECSC	EPI	MC	ECSC	EPI	
CTC	59.9	**59.9**	**80.3**	46.8	**52.7**	72.3	**62.0**
FA	**59.7**	60.3	80.7	**46.7**	**52.7**	**72.2**	62.1

Table 4.13: SubER scores (↓) on MuST-Cinema test set (MC), EC Short Clips (ECSC), and EuroParl Interviews (EPI) when the CTC-based audio segmentation (CTC) or the forced aligner (FA) method is used to extract the source-side timestamps.

background noise from EC Short Clips and EuroParl Interviews, as mentioned in §4.2.2.5.3. Table 4.14 shows the impact of background noise on the resulting subtitling quality. By comparing *1.* and *3.*, we notice that the presence of background noise causes an overall relative error increase of ∼5% on average over the two test sets and two language pairs. The degradation is caused both by the lower quality of the audio segmentation of SHAS and by worse outputs produced by the direct ST system, as the absence of noise during segmentation (*2.*) improves by an average of 1.7 SubER the results obtained without noise removal (*3.*). Creating models robust to background noise, though, is a task *per se* (Seltzer et al., 2013; Li et al., 2014; Mitra et al., 2017) and goes beyond the scope of this work.

Noise Removed	en-de		en-es		Avg.
	ECSC	EPI	ECSC	EPI	
1. Yes	59.9	80.3	66.3	72.3	52.7
2. Only Segm.	61.4	82.0	68.4	73.9	56.4
3. No	63.1	81.7	69.5	75.3	58.1

Table 4.14: SubER scores (↓) on EC Short Clips (ECSC) and EuroParl Interviews (EPI) with background noise removal for: both the audio segmentation with SHAS and the prediction of the direct ST system (*1.*); only the audio segmentation, but the noisy audio is fed as input to the direct ST model (*2.*); no noise removal (*3.*).

Chapter 5

Conclusions

5.1 Summary of contributions

This PhD thesis represents a comprehensive exploration of the multifaceted realm of speech translation, with a particular focus on two main aspects: simultaneous speech translation and automatic subtitling. My academic journey has unfolded across the inherent difficulties of integrating into standard ST systems the additional constraints required by these specific application scenarios, mainly regarding spatio-temporal aspects. During this journey, in particular, I initially focused on the direct or end-to-end architectures (Chapter 2) capable of directly generating the desired output from the input speech without intermediate steps, as they represented the emerging architectures when this PhD started in 2020.

Subsequently, the endeavor to enhance simultaneous speech translation (Chapter 3) and leverage existing ST systems without the need for extensive retraining or task-specific adaptation prompted an investigation into harnessing the intrinsic knowledge acquired by these systems during standard (offline) training for guiding simultaneous inference. This pursuit challenges a well-established paradigm of creating ad-hoc architectures for the task, advocating for a reassessment of the potential of applying offline direct ST models "as is" in the simultaneous scenario. Along this direction, my contributions can be summarized in the following findings:

- Offline-trained ST systems that are used in simultaneous inference can attain quality and latency that are competitive or even superior to those specifically trained for simultaneous processing;

- The intrinsic knowledge acquired by an offline-trained ST model, especially the

cross-attention information, can be directly leveraged for SimulST, resulting in low latency translations with reduced computational costs;

- Leveraging cross-attention information to extract alignment between speech and translation and using it as guidance for simultaneous inference not only provides a straightforward formulation but also achieves an optimal balance between quality and latency.

The chapter dedicated to automatic subtitling (Chapter 4) delves into the complexities of spatio-temporal constraints, unraveling the complicated interplay between text length, display timing, and user cognition. Here, I recognized the significance of prosody, pauses, and speech cues, shaping the development of direct architectures capable of directly accessing and exploiting these features. Specifically, my main findings in the automatic subtitling domain are as follows:

- To cope with data scarcity, direct multilingual multimodal models, which utilize both audio and textual cues to identify optimal segmentation points, revealed their effectiveness in automatic subtitle segmentation, delivering performance comparable to gold segmentation;

- Direct ST models demonstrate the capability of generating subtitles, which consist of segmented translations with corresponding timestamps, showing competitive performance against existing production tools.

As we approach the conclusion of this PhD, the insights gained and contributions made underscore the significant milestones achieved throughout my journey in the field of speech translation in the presence of specific constraints. Despite the challenges faced, these studies have been immensely rewarding, providing me with a deeper understanding of the obstacles and potential advancements within the dynamic domains of simultaneous communication and automatic subtitling.

Looking ahead, the broader goal of facilitating, supporting, and enriching the accessibility and comprehension of audiovisual materials by overcoming language barriers still persists: a lot has been done, but there is still room for many research directions. In the next section (Section 5.2), I therefore outline a collection of promising ideas for future research that I could not pursue within the three-year span of my journey. My hope is that these ideas will serve as a source of inspiration for those who, after me, will start their studies in the field of speech translation. The rapid evolution of the domain and these unrealized possibilities stand as invitations for further exploration

and innovation, encouraging the next generation of researchers to delve into the exciting challenges and opportunities that lie ahead.

As a concluding note in the final section of this chapter (Section 5.3), I elaborate on how simultaneous speech translation and automatic subtitling can be combined together, giving rise to *live subtitling*. Specifically, I trace the evolution in the field from the early stages of my PhD journey to the present, offering insights into the performance achievable through the innovations developed in these domains over the past three years.

5.2 Future Directions

5.2.1 Simultaneus Speech Translation

Advancing from Simultaneous to Streaming ST. The transition from simultaneous to streaming speech translation represents a fundamental step in addressing real-world applications. While simultaneous ST involves near-instantaneous conversion of speech to textual translation, streaming extends this concept further. In streaming scenarios, the system processes input in a continuous, real-time manner, catering to evolving content. In practice, the audio is not pre-segmented (as in simultaneous or even offline ST) but is represented by a single stream, requiring the system to dynamically determine what information from the past has to be retained in memory and what is no longer relevant for generating the current output. This evolution brings forth unique challenges in maintaining low latency, and ensuring seamless transitions without affecting the overall user experience. Investigating methodologies to optimize and adapt existing simultaneous decision policies for streaming applications is a crucial avenue for future exploration. Preliminary works exist on streaming ST (Iranzo-Sánchez et al., 2021; Iranzo Sanchez et al., 2022), but these studies predominantly utilize cascade systems as their backbone architecture. Consequently, an interesting research direction lies in investigating the integration of direct models into streaming ST frameworks and evaluating their performance in comparison to cascade systems.

Integrating External Knowledge for Enhanced Translation Quality and Accuracy. One promising direction, often referred to as knowledge injection (Nguyen et al., 2020a; Borghesi et al., 2020; Magnini et al., 2023), involves the integration of external sources of information, such as vocabularies. In the translation field, this technique aims to enhance translation quality in specific scenarios where named entities

and domain terminology are more frequent, as in live sports broadcasts and political debates, where current ST systems have been shown to struggle (Gaido et al., 2023b). By incorporating domain-specific lexicons or terminologies, the system gains contextual awareness, resulting in improved accuracy and coherence (Li et al., 2013; Dougal and Lonsdale, 2020; Gaido et al., 2021d, 2022b, 2023a). Research efforts should focus on developing robust methodologies for seamlessly injecting external knowledge but with the intrinsic additional difficulty of simultaneous speech translation, thus minimizing the required latency.

Latency Metrics with Real Audio-Translation Alignment. The accurate measurement of latency in the context of simultaneous ST is pivotal for assessing system performance. SimulST metrics should avoid approximations related to audio-translation alignment, an assumption underlying all the evaluation metrics proposed so far (Ma et al., 2020a; Papi et al., 2022a; Kano et al., 2023), ensuring a more precise and reliable evaluation of latency. This involves developing metrics that exactly detect the time delay between the spoken word in the source language and the corresponding translated output word (e.g., through audio-translation alignment obtained by external pre-trained models), providing a comprehensive understanding of the latency performance of the SimulST systems, taking also into account factors like pauses, hesitations, and other elements inherent in natural speech.

Leveraging Foundation Models to enhance Multilinguality and Performance. The incorporation of large multilingual models (LMMs), such as Whisper (Radford et al., 2023) and SeamlessM4T (Barrault et al., 2023), into the ST landscape holds the promise of multilingual benefits and improved overall performance. However, adapting LMMs for use in simultaneous translation poses considerable challenges. Overcoming the high latency[1] introduced by the typically large size of these models is a key hurdle (Macháček et al., 2023b). Finding innovative approaches to efficiently integrate or modify them for the simultaneous scenario without compromising their high quality and multilingual capabilities but with the stringent requirements of low-latency generation presents an exciting yet challenging research frontier.

[1]Estimated to be of approximately 3.3 seconds for the Whisper large-v2 model.

5.2.2 Automatic Subtitling

Direct Timestamp Generation Without Intermediate Transcription. A paradigm shift in subtitling involves eliminating the transcription step, directly generating subtitles with associated timestamps without intermediate representations, hence realizing the first direct system for "full" automatic subtitling. This direct generation of timestamps has the potential to reduce complexity, avoid error propagation, accelerate the subtitling process (with benefits in terms of latency), and open the application to source languages without written form (Lee et al., 2022). Practically, it would entail completing a full transition from models reliant on transcripts for all automatic subtitling subtasks (i.e., translation, segmentation into subtitles, and timestamp estimation) to more robust and efficient transcription-free models. Investigating methods to seamlessly integrate direct timestamp generation into the subtitling model is, therefore, essential for enhancing efficiency and accuracy.

Tailoring Translations for Specific Audience. Customizing translation approaches based on user characteristics such as cultural background and age introduces a layer of personalization to the subtitling process. For instance, adapting translations to be simpler for children can significantly enhance comprehension and engagement for this particular user group (Huang and Eskey, 1999; Capodieci et al., 2020). This approach also holds true for non-native speakers attempting to learn the language of the subtitles (Bisson et al., 2014). Investigating methods to dynamically adjust translation styles based on user profiles and demographics is a useful and interesting research direction, to align subtitling content with diverse audience needs and broadening its application.

Towards more Compressed and Time-Compliant Subtitles. The first IWSLT Evaluation Campaign on automatic subtitling in 2023 has revealed that all approaches, either exploiting cascade or direct ST systems, struggle in producing subtitles conforming to reading speed constraints (Agarwal et al., 2023), as measured with the CPS metric. This indicates an opportunity for enhancement in the production of more time-compliant subtitles. The challenge lies in refining existing models to produce translations with optimal CPS rates, ensuring more compressed yet informative and linguistically correct subtitles (Burnham et al., 2008; Szarkowska et al., 2016) and represents an interesting avenue for further research.

Introducing Timestamp Metric: Quantifying Temporal Precision. Timestamp accuracy is a critical aspect of subtitle quality. However, in previous research (Matusov et al., 2019, 2020; Koponen et al., 2020; Papi et al., 2023b), this aspect is rarely explored since no metric specifically assesses the quality of timestamp predictions. Proposing and adopting a metric explicitly designed to measure temporal precision would represent an important step to ensure a more comprehensive evaluation. This dedicated metric should consider factors such as synchronization with speech, and precision in reflecting the actual translation content. Moreover, developing a standardized evaluation framework would provide a valuable tool for researchers and practitioners to assess and compare different subtitle generation models also under the temporal aspects.

Adapting Foundation Models for Multilingual Subtitling. As also discussed for simultaneous ST, leveraging large multilingual models (LMMs) for multilingual subtitling presents an opportunity to enhance performance and language diversity as well as to meet the growing demand for subtitles. Adapting existing LMMs for subtitling tasks should involve optimization, for instance through fine-tuning, to generate more compressed translations formatted into lines and blocks, thus with the emission of the required `<eob>` and `<eol>` markers. Future research should delve into a seamless integration of these characteristics into current LMM architectures while ensuring the retention of their multilingual capabilities and preserving or even improving performance.

5.3 The Evolution of Live Subtitling

In the culmination of this comprehensive exploration of Simultaneous Speech Translation and Automatic Subtitling, this concluding section unfolds the potential convergence of these two research areas into a unified task: Live Subtitling. Live subtitling (Aliprandi et al., 2014) entails generating an incremental translation from partial input speech (Chapter 3) while adhering to the spatio-temporal constraints typical of subtitling (Chapter 4).

Initially designed for intralingual subtitles – those in the same language as the source speech – to assist the deaf or hard-of-hearing in following live TV programs (Lambourne, 2006), live subtitling expanded its scope to include interlingual subtitles Dawson (2019) – those in a language different from the source speech.

In its interlingual mode, live subtitling is an emerging practice, prompting the industry to experiment with different profiles for the role of interlingual live subtitlers

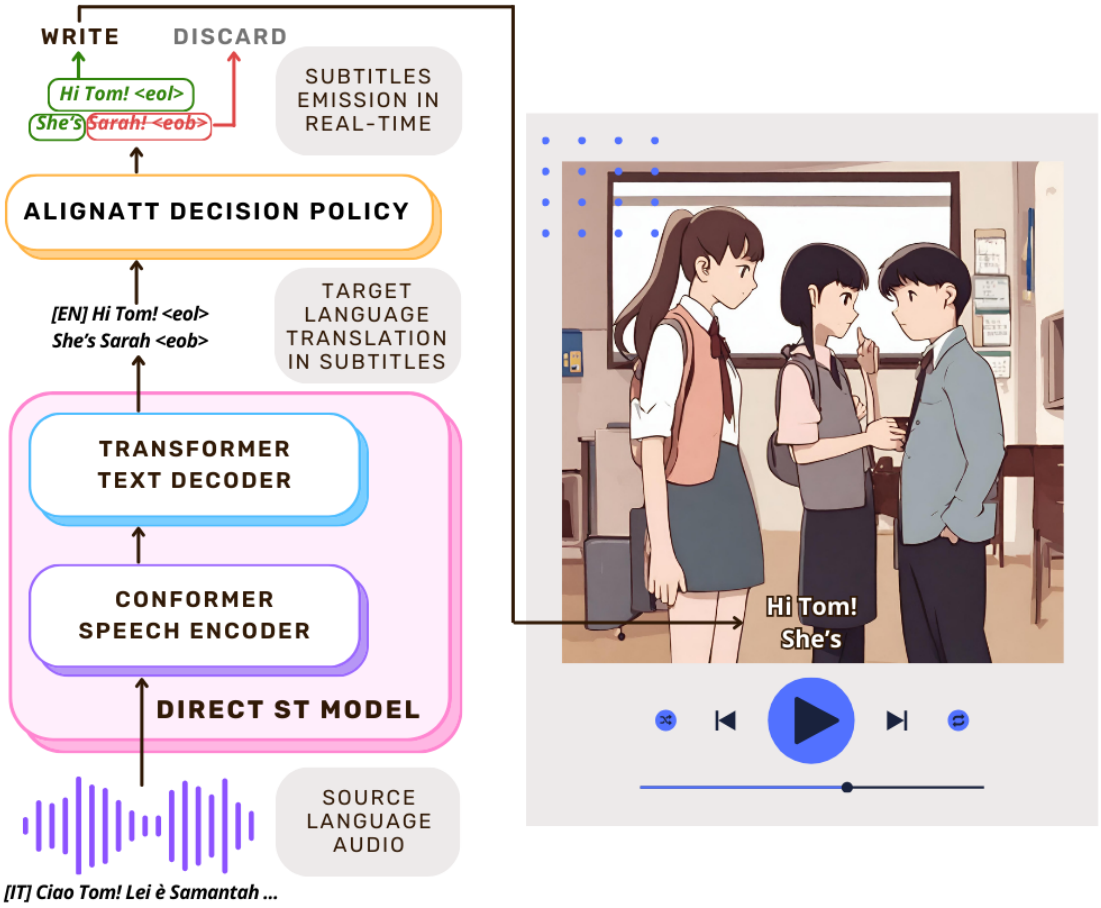


Figure 5.1: Architecture of the offline-trained direct model for automatic subtitling enhanced with the `AlignAtt` decision policy for simultaneous inference.

(Pöchhacker and Remael, 2020; Fantinuoli et al., 2021). This role demands skills from three disciplines: subtitling, respeaking (Lambourne, 2006), and simultaneous interpreting (Marsh, 2004). Consequently, the availability of highly skilled professionals for interlingual live subtitling falls short of meeting the growing demands in real-time multilingual communication.

In the early stages of my research studies, I attempted to realize the first live subtitling system based on a direct architecture (Karakanta et al., 2021b). At that time, the technologies were in their initial development, and I employed the wait-k decision policy, a straightforward yet simple strategy, to enable a subtitling system to operate in simultaneous scenarios (Section 3.1.1.1). Moreover, the system was built specifically for the live subtitling task, with the wait-k policy employed during training, using a specific value of k.

After two years of research, assessing the current evolution status of live subtitling

systems no longer necessitates the creation of ad-hoc models for the task. Instead, more advanced solutions can exploit the same offline systems trained at that time, enhanced with the `AlignAtt`[2] decision policy proposed in my latest work on simultaneous ST (Section 3.2.3), to build a live subtitling system without the need for any adaptations. The complete architecture is depicted in Figure 5.1.

As illustrated in the example, the audio speech in Italian (IT: "Ciao Tom! Lei è Samantah", EN: "Hi Tom! She is Samantah") is processed by an offline-trained direct ST model for automatic subtitling (i.e., able to generate the `<eob>` and `<eol>` markers) composed of a Conformer speech encoder and a Transformer text decoder. Subsequently, the model predicts the subtitle-segmented translation into English (EN: "Hi Tom! `<eol>` She's Sarah `<eob>`"), which is then processed by the `AlignAtt` policy, which determines the words to be emitted (WRITE: "Hi Tom! `<eol>` She's") and those to be discarded (DISCARD: "Sarah! `<eob>`"). Lastly, the selected translation text is shown on the screen in subtitle format (with markers substituted with newlines).

Figure 5.2 shows the simultaneous results in terms of quality (BLEU) and latency (LAAL) in the three languages (en-de, en-it, en-fr) analyzed in previous work, and, in Table 5.1, the detailed scores are provided together with the corresponding CPL conformity.

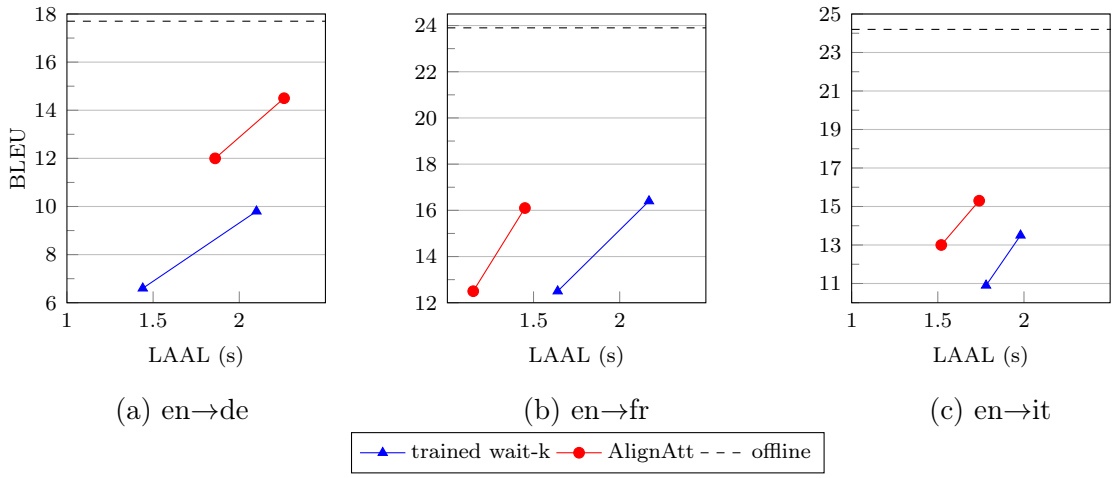
It can be observed that the performance of the offline-trained models applying `AlignAtt` at inference time is noticeably superior to that of the systems trained with the wait-k policy. The `AlignAtt` curves (in red, round dots) always stay above (i.e., better – higher – quality) or towards the left (i.e., better – lower – latency) with respect to the wait-k curves (in blue, triangle dots). Also, the CPL conformity is higher with the `AlignAtt` policy, with values approaching the offline ones. These results demonstrate that the technological advancements pursued during my PhD allow for outperforming the results published in 2021, even using the same model without any re-training or adaptation.

As an additional proof of concept, I used the offline ST model developed for my most recent work on automatic subtitling (Section 4.2.2), and with which I participated in the first IWSLT Evaluation Campaign on automatic subtitling (Papi et al., 2023b), applying `AlignAtt` to obtain live subtitling outputs. This en-de model[3] is a better performing direct architecture trained on high-resource conditions.[4] The simultaneous

[2]We set the simultaneous hyper-parameter frame (f) to 2 and 20.

[3]All the material of the FBK submission at IWSLT is publicly accessible at: `https://github.com/hlt-mt/FBK-fairseq/blob/master/fbk_works/IWSLT_2023.md`.

[4]All the datasets available for the constrained data condition (`https://iwslt.org/2023/subtitling`) were used for training the offline ST model.



(a) en→de (b) en→fr (c) en→it

Figure 5.2: LAAL-BLEU curves for three language pairs of MuST-Cinema. Scores are obtained with SimulEval v1.1.0.

Model	**en-it**			**en-de**			**en-fr**		
	BLEU	LAAL	CPL	BLEU	LAAL	CPL	BLEU	LAAL	CPL
offline	24.2	-	93.1	17.7	-	95.0	23.9	-	95.3
wait-k ($k = 3$)	10.9	1781	90.5	6.6	1438	93.6	12.5	1639	91.4
wait-k ($k = 5$)	13.5	1977	91.2	9.8	2100	90.1	16.4	2173	93.6
`AlignAtt` ($f = 2$)	13.0	1519	92.3	12.0	1861	95.0	12.5	1146	94.7
`AlignAtt` ($f = 20$)	15.3	1738	92.6	14.5	2260	95.4	16.1	1449	93.7

Table 5.1: BLEU without `<eol>` and `<eob>` (sacreBLEU v2.3.1), LAAL (in milliseconds) and CPL conformity (%) on three language pairs (en-de, en-fr, en-it) of MuST-Cinema amara.

results are shown in Figure 5.3, and the numeric BLEU/LAAL values, together with CPL conformity, are presented in Table 5.2.

As it can be seen in the first row of Table 5.2, the translation quality of the offline model increased by 8 BLEU points compared to the system developed in 2021 (Table 5.1). This huge performance gain is yielded by two factors. On one side, by a more advanced architecture since Transformer was replaced by the more competitive Conformer. On the other side, by the use of more training data, even if most of them were synthetically segmented using our multilingual multimodal subtitle segmenter (Section 4.2.1), which also explains a slight drop in terms of CPL conformity compliance (-2%). In Figure 5.3, it can be observed that the increase in translation quality of the offline system is also reflected by improved simultaneous performance, showing an average of 8 BLEU points gain with almost no additional latency compared to the 2021 systems (Figure 5.2).

In bringing together the realms of simultaneous ST and automatic subtitling, this

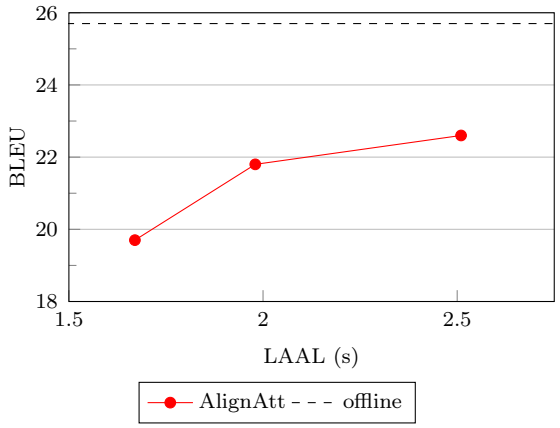


Figure 5.3: LAAL-BLEU curves on MuST-Cinema amara en-de. Scores are obtained with SimulEval v1.1.0.

Model	BLEU	LAAL	CPL
offline	25.7	-	91.0
`AlignAtt` ($f = 2$)	19.7	1666	88.3
`AlignAtt` ($f = 6$)	21.8	1975	88.2
`AlignAtt` ($f = 12$)	22.6	2508	88.2

Table 5.2: BLEU without `<eol>` and `<eob>` (sacreBLEU v2.3.1), LAAL (in milliseconds) and CPL conformity (%) on MuST-Cinema amara en-de.

final section unfolded a brief exploration, throughout empirical experiments, of live subtitling models, providing interesting insights into their dynamics and performance in delivering simultaneous subtitles. The promising potential observed in these automatic models, marked by escalating quality and reduced latency, holds substantial implications for translators and interpreters. By alleviating their workload, these tools emerge as possible valuable aids (Fantinuoli and Dastyar, 2022; Fantinuoli, 2023), fostering a more streamlined and efficient human-computer collaboration in language-related tasks (Prandi, 2015, 2020) and representing a valid direction for future research endeavors.

Such a conclusion goes well beyond my expectations at the beginning of this long and rewarding PhD journey. Along the way, I dedicated myself to achieving substantial advancements in these two challenging application domains of speech translation and, by combining insights derived from three years of research, I hope to have laid a foundation for future endeavors in this field.

Bibliography

Vìctor M. Garro Abarca, Pedro R. Palos-Sanchez, and Enrique Rus-Arias. Working in virtual teams: A systematic literature review and a bibliometric analysis. *IEEE Access*, 8:168923–168940, 2020. doi: 10.1109/ACCESS.2020.3023546.

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Kevin Duh, Yannick Estève, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutai Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Kr. Ojha, John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.1. URL `https://aclanthology.org/2023.iwslt-1.1`.

Carlo Aliprandi, Cristina Scudellari, Isabella Gallucci, Nicola Piccinini, Matteo Raffaelli, Arantza Pozo, Aitor Álvarez, Haritz Arzelus, Renato Cassaca, Tiago Luis, João Neto, Carlos Mendes, Sérgio Paulo, and Marcio Viveiros. Automatic live subtitling: state of the art, expectations and current trends. In *Proceedings of NAB Broadcast Engineering Conference: Papers on Advanced Media Technologies*, volume 13, Las Vegas, 04 2014. doi: 10.13140/RG.2.1.3995.3440.

Aitor Álvarez, Carlos Mendes, Matteo Raffaelli, Tiago Luís, Sérgio Paulo, Nicola Piccinini, Haritz Arzelus, João Neto, Carlo Aliprandi, and Arantza Pozo. Automating live and batch subtitling of multimedia contents for several european languages. *Multimedia Tools and Applications*, 75:1–31, 07 2015. doi: 10.1007/s11042-015-2794-z.

Aitor Álvarez, Marina Balenciaga, Arantza del Pozo, Haritz Arzelus, Anna Matamala, and Carlos-D. Martínez-Hinarejos. Impact of automatic segmentation on the quality, productivity and self-reported post-editing effort of intralingual subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3049–3053, Portorož, Slovenia, May 2016. URL `https://aclanthology.org/L16-1487`.

Aitor Álvarez, Carlos-D. Martínez-Hinarejos, Haritz Arzelus, Marina Balenciaga, and Arantza del Pozo. Improving the automatic segmentation of subtitles through conditional random field. *Speech Communication*, 88:83–95, 2017. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2017.01.010. URL `https://www.sciencedirect.com/science/article/pii/S0167639316300127`.

Antonios Anastasopoulos and David Chiang. A case study on using speech-to-translation alignments for language documentation. In Antti Arppe, Jeff Good, Mans Hulden, Jordan Lachler, Alexis Palmer, and Lane Schwartz, editors, *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 170–178, Honolulu, March 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-0123. URL `https://aclanthology.org/W17-0123`.

Antonios Anastasopoulos and David Chiang. Tied Multitask Learning for Neural Speech Translation. 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 82–91, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1008. URL `https://www.aclweb.org/anthology/N18-1008`.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok,

Thailand (online), August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwslt-1.1. URL https://aclanthology.org/2021.iwslt-1.1.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online), May 2022. doi: 10.18653/v1/2022.iwslt-1.10. URL https://aclanthology.org/2022.iwslt-1.10.

Andrei Andrusenko, Rauf Nasretdinov, and Aleksei Romanenko. Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition. *arXiv preprint arXiv:2208.07657*, 2022.

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. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online, July 2020. doi: 10.18653/v1/2020.iwslt-1.1. URL https://aclanthology.org/2020.iwslt-1.1.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France, May 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.520.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy, July 2019. doi: 10.18653/v1/P19-1126. URL https://aclanthology.org/P19-1126.

Naveen Arivazhagan, Colin Cherry, Isabelle Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 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, 2020.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July 2018. doi: 10.18653/v1/P18-1073. URL https://aclanthology.org/P18-1073.

Wilker Aziz, Sheila C. M. de Sousa, and Lucia Specia. Cross-lingual sentence compression for subtitles. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 103–110, Trento, Italy, May 28–30 2012. URL https://aclanthology.org/2012.eamt-1.33.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. A Comparative Study on End-to-end Speech to Text Translation. In *Proceedings of International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 792–799, Sentosa, Singapore, December 2019.

Parnia Bahar, Patrick Wilken, Mattia A. Di Gangi, and Evgeny Matusov. Without further ado: Direct and simultaneous speech translation by AppTek in 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 52–63, Bangkok, Thailand (online), August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwslt-1.5. URL https://aclanthology.org/2021.iwslt-1.5.

Parnia Bahar, Patrick Wilken, Javier Iranzo-Sánchez, Mattia Di Gangi, Evgeny Matusov, and Zoltán Tüske. Speech translation with style: AppTek's submissions to the IWSLT subtitling and formality tracks in 2023. In Elizabeth Salesky, Marcello Federico,

and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 251–260, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.22. URL https://aclanthology.org/2023.iwslt-1.22.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949, 2016. doi: 10.1109/ICASSP.2016.7472618.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. URL https://aclanthology.org/W05-0909.

Shahana Bano, Pavuluri Jithendra, Gorsa Lakshmi Niharika, and Yalavarthi Sikhi. Speech to text translation enabling multilingualism. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–4, 2020. doi: 10.1109/INOCON50539.2020.9298280.

Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. Towards speech-to-text translation without speech recognition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 474–479, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2076.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. Low-Resource Speech-to-Text Translation. In *Proc. Interspeech 2018*, pages 1298–1302, 2018. doi: 10.21437/Interspeech.2018-1326.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. Pre-training on High-resource Speech Recognition Improves Low-resource Speech-to-text Translation. In *Proceedings of the 2019 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 58–68, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1006. URL https://www.aclweb.org/anthology/N19-1006.

Henri C. Barik. Simultaneous interpretation: Qualitative and linguistic data. *Language and Speech*, 18(3), 1975. doi: 10.1177/002383097501800310.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, et al. Seamlessm4t-massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*, 2023.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 1304–1313, New Orleans, Louisiana, June 2018. doi: 10.18653/v1/N18-1118. URL https://aclanthology.org/N18-1118.

Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1–3):177–210, feb 1999. ISSN 0885-6125. doi: 10.1023/A:1007506220214. URL https://doi.org/10.1023/A:1007506220214.

Maximiliana Behnke and Kenneth Heafield. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online, November 2020. doi: 10.18653/v1/2020.emnlp-main.211. URL https://aclanthology.org/2020.emnlp-main.211.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 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 IJCNLP*, pages 2873–2887, Online, August 2021. doi: 10.18653/v1/2021.acl-long.224. URL https://aclanthology.org/2021.acl-long.224.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-End Automatic Speech Translation of Audiobooks. In *Proceedings of ICASSP*

2018 - IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Alberta, Canada, April 2018.

Laurent Besacier, Bowen Zhou, and Yuqing Gao. Towards speech translation of non written languages. In *2006 IEEE Spoken Language Technology Workshop*, pages 222–225, 2006. doi: 10.1109/SLT.2006.326795.

Marie-Josée Bisson, Walter J. B. Van Heuven, Kathy Conklin, and Richard J. Tunney. Processing of native and foreign language subtitles in films: An eye tracking study. *Applied Psycholinguistics*, 35(2):399–418, 2014. doi: 10.1017/S0142716412000434.

Alan W. Black, Ralf D. Brown, Robert Frederking, Kevin Lenzo, John Moody, Alexander Rudnicky, Rita Singh, and Eric Steinbrecher. Rapid Devlopement Of Speech-To-Speech Translation Systems. In *Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002*, Denver, Colorado, September 2002.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 06 2017. ISSN 2307-387X. doi: 10.1162/tacl_a_00051. URL `https://doi.org/10.1162/tacl_a_00051`.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. doi: 10.3115/v1/W14-3302. URL `https://aclanthology.org/W14-3302`.

Ondřej Bojar, Dominik Macháček, Sangeet Sagar, Otakar Smrž, Jonáš Kratochvíl, Peter Polák, Ebrahim Ansari, Mohammad Mahmoudi, Rishu Kumar, Dario Franceschini, Chiara Canton, Ivan Simonini, Thai-Son Nguyen, Felix Schneider, Sebastian Stüker, Alex Waibel, Barry Haddow, Rico Sennrich, and Philip Williams. ELITR multilingual live subtitling: Demo and strategy. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 271–277, Online, April 2021. doi: 10.18653/v1/2021.eacl-demos.32. URL `https://aclanthology.org/2021.eacl-demos.32`.

Andrea Borghesi, Federico Baldo, Michele Lombardi, and Michela Milano. Injective domain knowledge in neural networks for transprecision computing. In Giuseppe Nicosia, Varun Ojha, Emanuele La Malfa, Giorgio Jansen, Vincenzo Sciacca, Panos Pardalos, Giovanni Giuffrida, and Renato Umeton, editors, *Machine Learning, Optimization, and Data Science*, pages 587–600, Cham, 2020. Springer International Publishing. ISBN 978-3-030-64583-0.

François Buet and François Yvon. Toward Genre Adapted Closed Captioning. In *Proc. Interspeech 2021*, pages 4403–4407, 2021. doi: 10.21437/Interspeech.2021-1762.

Maxime Burchi and Valentin Vielzeuf. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15, 2021. doi: 10.1109/ASRU51503.2021.9687874.

Denis Burnham, Greg Leigh, William Noble, Caroline Jones, Michael Tyler, Leonid Grebennikov, and Alex Varley. Parameters in Television Captioning for Deaf and Hard-of-Hearing Adults: Effects of Caption Rate Versus Text Reduction on Comprehension. *The Journal of Deaf Studies and Deaf Education*, 13(3):391–404, 03 2008. ISSN 1081-4159. doi: 10.1093/deafed/enn003. URL `https://doi.org/10.1093/deafed/enn003`.

Lindsay Bywood, Martin Volk, Mark Fishel, and Panayota Georgakopoulou. Parallel subtitle corpora and their applications in machine translation and translatology. *Perspectives*, 21(4):595–610, 2013. doi: 10.1080/0907676X.2013.831920. URL `https://doi.org/10.1080/0907676X.2013.831920`.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 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 2016.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of Bleu in machine translation research. In Diana McCarthy and Shuly Wintner, editors, *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy, April 2006. Association for Computational Linguistics. URL `https://aclanthology.org/E06-1032`.

Agnese Capodieci, Cesare Cornoldi, Elizabeth Doerr, Laura Bertolo, and Barbara Carretti. The use of new technologies for improving reading comprehension. *Frontiers in Psychology*, 11, 2020. ISSN 1664-1078. doi: 10.3389/fpsyg.2020.00751. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2020.00751.

Mary Carroll and Jan Ivarsson. *Code of Good Subtitling Practice*. Simrishamn: TransEdit, 1998. URL http://esist.org/wp-content/uploads/2016/06/Code-of-Good-Subtitling-Practice.PDF.pdf.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155, 2021. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2020.101155. URL https://www.sciencedirect.com/science/article/pii/S0885230820300887.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964, 2016. doi: 10.1109/ICASSP.2016.7472621.

Chih-Chiang Chang and Hung-Yi Lee. Exploring Continuous Integrate-and-Fire for Adaptive Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 5175–5179, 2022. doi: 10.21437/Interspeech.2022-10627.

Xuankai Chang, Aswin Shanmugam Subramanian, Pengcheng Guo, Shinji Watanabe, Yuya Fujita, and Motoi Omachi. End-to-end asr with adaptive span self-attention. In *INTERSPEECH*, 2020.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 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 2021. doi: 10.18653/v1/2021.findings-acl.406. URL https://aclanthology.org/2021.findings-acl.406.

Yi-Ting Chen and Ming-Chou Ho. Eye movement patterns differ while watching captioned videos of second language vs. mathematics lessons. *Learning and Individual Differences*, 93:102106, 2022. ISSN 1041-6080. doi: https://doi.org/10.1016/j.lindif.2021.102106. URL https://www.sciencedirect.com/science/article/pii/S1041608021001436.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online, November 2020. doi: 10.18653/v1/2020.emnlp-main.42. URL `https://aclanthology.org/2020.emnlp-main.42`.

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

Colin Cherry, Naveen Arivazhagan, Dirk Padfield, and Maxim Krikun. Subtitle Translation as Markup Translation. In *Proc. Interspeech 2021*, pages 2237–2241, 2021. doi: 10.21437/Interspeech.2021-744.

A. Chmiel, A. Szarkowska, Danijel Korzinek, Agnieszka Lijewska, Łukasz Dutka, Łukasz Brocki, and K. Marasek. Ear–voice span and pauses in intra- and interlingual respeaking: An exploratory study into temporal aspects of the respeaking process. *Applied Psycholinguistics*, 38:1201 – 1227, 2017.

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

Jorge Díaz Cintas and Aline Remael. *Subtitling: Concepts and Practices*. Translation practices explained. Routledge, 2021.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. doi: 10.18653/v1/W19-4828. URL `https://aclanthology.org/W19-4828`.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. doi: 10.18653/v1/2020.acl-main.747. URL `https://aclanthology.org/2020.acl-main.747`.

Marta R Costa-jussà, James Cross, Onur Çelebi, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A survey of multilingual neural machine translation. 53(5), sep 2020. ISSN 0360-0300. doi: 10.1145/3406095. URL https://doi.org/10.1145/3406095.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 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 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2079. URL https://aclanthology.org/N18-2079.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 933–941. JMLR.org, 2017.

Hayley Dawson. Feasibility, quality and assessment of interlingual live subtitling: A pilot study. *Journal of Audiovisual Translation*, 2(2):36–56, Dec. 2019. doi: 10.47476/jat.v2i2.72. URL https://jatjournal.org/index.php/jat/article/view/72.

Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 1746–1750, 2022. doi: 10.21437/Interspeech.2022-933.

Pan Deng, Shihao Chen, Weitai Zhang, Jie Zhang, and Lirong Dai. The USTC's dialect speech translation system for IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 102–112, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.5. URL https://aclanthology.org/2023.iwslt-1.5.

Tracey M. Derwing and Murray J. Munro. Putting accent in its place: Rethinking obstacles to communication. *Language Teaching*, 42(4):476–490, 2009. doi: 10.1017/S026144480800551X.

Florian Dessloch, Thanh-Le Ha, Markus Müller, Jan Niehues, Thai-Son Nguyen, Ngoc-Quan Pham, Elizabeth Salesky, Matthias Sperber, Sebastian Stüker, Thomas Zenkel, and Alexander Waibel. KIT lecture translator: Multilingual speech translation with one-shot learning. In Dongyan Zhao, editor, *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 89–93, Santa Fe, New Mexico, August 2018. Association for Computational Linguistics. URL `https://aclanthology.org/C18-2020`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. MuST-C: a Multilingual Speech Translation Corpus. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1202. URL `https://aclanthology.org/N19-1202`.

Mattia A. Di Gangi, Matteo Negri, Viet Nhat Nguyen, Amirhossein Tebbifakhr, and Marco Turchi. Data augmentation for end-to-end speech translation: FBK@IWSLT '19. In Jan Niehues, Rolando Cattoni, Sebastian Stüker, Matteo Negri, Marco Turchi, Thanh-Le Ha, Elizabeth Salesky, Ramon Sanabria, Loic Barrault, Lucia Specia, and Marcello Federico, editors, *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, November 2-3 2019b. Association for Computational Linguistics. URL `https://aclanthology.org/2019.iwslt-1.14`.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. Adapting Transformer to End-to-End Spoken Language Translation. In *Proc. Interspeech 2019*, pages 1133–1137, 2019c. doi: 10.21437/Interspeech.2019-3045.

Mattia A. Di Gangi, Marco Gaido, Matteo Negri, and Marco Turchi. On Target Segmentation for Direct Speech Translation. In *Proceedings of the 14th Conference*

of the Association for Machine Translation in the Americas (AMTA 2020), pages 137–150, Virtual, October 2020.

Linhao Dong and Bo Xu. Cif: Continuous integrate-and-fire for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083, 2020. doi: 10.1109/ICASSP40776.2020.9054250.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018. doi: 10.1109/ICASSP.2018.8462506.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. Learning when to translate for streaming speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.50. URL https://aclanthology.org/2022.acl-long.50.

Duane K. Dougal and Deryle Lonsdale. Improving NMT quality using terminology injection. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4820–4827, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.593.

Yichao Du, Zhirui Zhang, Weizhi Wang, Boxing Chen, Jun Xie, and Tong Xu. Regularizing end-to-end speech translation with triangular decomposition agreement. *Proc. of AAAI*, 36(10), Jun. 2022. doi: 10.1609/aaai.v36i10.21303.

Vernandi Dyzel, Rony Oosterom-Calo, Mijkje Worm, and Paula S. Sterkenburg. Assistive technology to promote communication and social interaction for people with deafblindness: A systematic review. *Frontiers in Education*, 5, 2020. ISSN 2504-284X. doi: 10.3389/feduc.2020.578389. URL https://www.frontiersin.org/articles/10.3389/feduc.2020.578389.

Johanes Effendi, Yogesh Virkar, Roberto Barra-Chicote, and Marcello Federico. Duration modeling of neural tts for automatic dubbing. *ICASSP 2022 - 2022 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8037–8041, 2022.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. Interspeech 2020*, pages 1461–1465, 2020. doi: 10.21437/Interspeech.2020-1241. URL `http://dx.doi.org/10.21437/Interspeech.2020-1241`.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. From bilingual to multilingual neural machine translation by incremental training. In Fernando Alva-Manchego, Eunsol Choi, and Daniel Khashabi, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy, July 2019. doi: 10.18653/v1/P19-2033. URL `https://aclanthology.org/P19-2033`.

Thierry Etchegoyhen, Lindsay Bywood, Mark Fishel, Panayota Georgakopoulou, Jie Jiang, Gerard van Loenhout, Arantza del Pozo, Mirjam Sepesy Maučec, Anja Turner, and Martin Volk. Machine translation for subtitling: A large-scale evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 46–53, Reykjavik, Iceland, May 2014. URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/463_Paper.pdf`.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. STEMM: Self-learning with speech-text manifold mixup for speech translation. In *Proc. ACL 2022*, Dublin, Ireland, May 2022. doi: 10.18653/v1/2022.acl-long.486.

Claudio Fantinuoli. Towards ai-enhanced computer-assisted interpreting. In *Interpreting Technologies – Current and Future Trends*, pages 46–71. John Benjamins, 2023. URL `https://www.jbe-platform.com/content/books/9789027249456-ivitra.37.03fan`.

Claudio Fantinuoli and Vorya Dastyar. Interpreting and the emerging augmented paradigm. *Interpreting and Society*, 2(2):185–194, 2022.

Claudio Fantinuoli and Maddalena Montecchio. Defining maximum acceptable latency of ai-enhanced cai tools. *arXiv preprint arXiv:2201.02792*, 2022.

Claudio Fantinuoli and Bianca Prandi. Towards the evaluation of automatic simultaneous speech translation from a communicative perspective. In Marcello Federico,

Alex Waibel, Marta R. Costa-jussà, Jan Niehues, Sebastian Stuker, and Elizabeth Salesky, editors, *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 245–254, Bangkok, Thailand (online), August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwslt-1.29. URL `https://aclanthology.org/2021.iwslt-1.29`.

Claudio Fantinuoli, Giulia Marchesini, David Landan, and Lukas Horak. Kudo interpreter assist: Automated real-time support for remote interpretation. *Translating and the Computer 43*, page 68, 2021.

Marcello Federico, Yogesh Virkar, Robert Enyedi, and Roberto Barra-Chicote. Evaluating and Optimizing Prosodic Alignment for Automatic Dubbing. In *Proc. Interspeech 2020*, pages 1481–1485, 2020. doi: 10.21437/Interspeech.2020-2983. URL `http://dx.doi.org/10.21437/Interspeech.2020-2983`.

Javier Ferrando, Gerard I Gállego, Belen Alastruey, Carlos Escolano, and Marta R Costa-jussà. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. *arXiv e-prints*, pages arXiv–2205, 2022.

Chris Fournier. Evaluating text segmentation using boundary edit distance. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1702–1712, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://aclanthology.org/P13-1167`.

Chris Fournier and Diana Inkpen. Segmentation similarity and agreement. In Eric Fosler-Lussier, Ellen Riloff, and Srinivas Bangalore, editors, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 152–161, Montréal, Canada, June 2012. Association for Computational Linguistics. URL `https://aclanthology.org/N12-1016`.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto

Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.2.

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

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

Ryo Fukuda, Yuka Ko, Yasumasa Kano, Kosuke Doi, Hirotaka Tokuyama, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. NAIST simultaneous speech-to-text translation system for IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 286–292, Dublin, Ireland (in-person and online), May 2022. doi: 10.18653/v1/2022.iwslt-1.25. URL https://aclanthology.org/2022.iwslt-1.25.

Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Yuka Ko, Tomoya Yanagita, Kosuke Doi, Mana Makinae, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. NAIST simultaneous speech-to-speech translation system for IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 330–340, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.31. URL https://aclanthology.org/2023.iwslt-1.31.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, Mauro Cettolo, and Marco Turchi. Contextualized Translation of Automatically Segmented Speech. In *Proceedings of Interspeech 2020*, pages 1471–1475, October 2020a. doi: 10.21437/Interspeech.2020-2860. URL http://dx.doi.org/10.21437/Interspeech.2020-2860.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. End-to-end speech-translation with knowledge distillation: FBK@IWSLT2020. In Marcello Federico, Alex Waibel, Kevin Knight, Satoshi Nakamura, Hermann Ney, Jan Niehues, Sebastian Stüker, Dekai Wu, Joseph Mariani, and Francois Yvon, editors, *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.iwslt-1.8. URL https://aclanthology.org/2020.iwslt-1.8.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. CTC-based compression for direct speech translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 690–696, Online, April 2021a. doi: 10.18653/v1/2021.eacl-main.57. URL https://aclanthology.org/2021.eacl-main.57.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. On Knowledge Distillation for Direct Speech Translation . In *Proceedings of CLiC-IT 2020*, Online, March 2021b. URL http://ceur-ws.org/Vol-2769/paper_28.pdf.

Marco Gaido, Matteo Negri, Mauro Cettolo, and Marco Turchi. Beyond voice activity detection: Hybrid audio segmentation for direct speech translation. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 55–62, Trento, Italy, 12–13 November 2021c. Association for Computational Linguistics. URL https://aclanthology.org/2021.icnlsp-1.7.

Marco Gaido, Susana Rodríguez, Matteo Negri, Luisa Bentivogli, and Marco Turchi. Is "moby dick" a whale or a bird? named entities and terminology in speech translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1707–1716, Online and Punta Cana, Dominican Republic, November 2021d. doi: 10.18653/v1/2021.emnlp-main.128. URL https://aclanthology.org/2021.emnlp-main.128.

Marco Gaido, Matteo Negri, and Marco Turchi. Direct speech-to-text translation models as students of text-to-text models. *Italian Journal of Computational Linguistics*, 2022a.

Marco Gaido, Matteo Negri, and Marco Turchi. Who are we talking about? handling person names in speech translation. In Elizabeth Salesky, Marcello Federico, and Marta Costa-jussà, editors, *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 62–73, Dublin, Ireland (in-person and online), May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.6. URL https://aclanthology.org/2022.iwslt-1.6.

Marco Gaido, Matteo Negri, and Marco Turchi. Direct speech-to-text translation models as students of text-to-text models. *IJCoL. Italian Journal of Computational Linguistics*, 8(8-1), 2022c.

Marco Gaido, Matteo Negri, and Marco Turchi. Direct speech-to-text translation models as students of text-to-text models. *IJCoL. Italian Journal of Computational Linguistics*, 8(8-1), 2022d.

Marco Gaido, Sara Papi, Dennis Fucci, Giuseppe Fiameni, Matteo Negri, and Marco Turchi. Efficient yet competitive speech translation: FBK@IWSLT2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 177–189, Dublin, Ireland (in-person and online), May 2022e. doi: 10.18653/v1/2022.iwslt-1.13. URL https://aclanthology.org/2022.iwslt-1.13.

Marco Gaido, Sara Papi, Matteo Negri, and Marco Turchi. Joint Speech Translation and Named Entity Recognition. In *Proc. INTERSPEECH 2023*, pages 47–51, 2023a. doi: 10.21437/Interspeech.2023-1767.

Marco Gaido, Yun Tang, Ilia Kulikov, Rongqing Huang, Hongyu Gong, and Hirofumi Inaguma. Named entity detection and injection for direct speech translation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023b. doi: 10.1109/ICASSP49357.2023.10094689.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. 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 4453–4462, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1453. URL https://aclanthology.org/D19-1453.

Panayota Georgakopoulou. Template files:: The holy grail of subtitling. *Journal of Audiovisual Translation*, 2(2):137–160, Dec. 2019. doi: 10.47476/jat.v2i2.84. URL https://www.jatjournal.org/index.php/jat/article/view/84.

Hongyu Gong, Yun Tang, Juan Pino, and Xian Li. Pay better attention to attention: Head selection in multilingual and multi-domain sequence modeling. *Advances in Neural Information Processing Systems*, 34:2668–2681, 2021.

Henrik Gottlieb. Subtitles and international anglification. *Nordic Journal of English Studies*, 3:219, 01 2004. doi: 10.35360/njes.32.

Edward Gow-Smith, Alexandre Berard, Marcely Zanon Boito, and Ioan Calapodescu. NAVER LABS Europe's multilingual speech translation systems for the IWSLT 2023

low-resource track. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 144–158, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.10. URL https://aclanthology.org/2023.iwslt-1.10.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1764–II–1772, 2014.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 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, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Roberto Gretter, Marco Matassoni, and Daniele Falavigna. Seed words based data selection for language model adaptation. *arXiv preprint arXiv:2107.09433*, 2021.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 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 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1099.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, Jing Shi, Shinji Watanabe, Kun Wei, Wangyou Zhang, and Yuekai Zhang. Recent

developments on espnet toolkit boosted by conformer. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878, 2021. doi: 10.1109/ICASSP39728.2021.9414858.

Steven Gutstein, Olac Fuentes, and Eric Freudenthal. Knowledge Transfer in Deep Convolutional Neural Nets. *International Journal on Artificial Intelligence Tools*, 17 (03):555–567, 2008. doi: 10.1142/S0218213008004059.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *Institute for Anthropomatics and Robotics*, 2(10.12):16, 2016.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. Learning shared semantic space for speech-to-text translation. In *Findings ACL-IJCNLP 2021*, Online, August 2021. doi: 10.18653/v1/2021.findings-acl.195.

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*, pages 62–68, Online, July 2020. doi: 10.18653/v1/2020.iwslt-1.5. URL `https://aclanthology.org/2020.iwslt-1.5`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning, 2020.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In Alexey Karpov, Oliver Jokisch, and Rodmonga Potapova, editors, *Speech and Computer*, pages 198–208, Cham, 2018. ISBN 978-3-319-99579-3.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. In *Proc. of NIPS Deep Learning and Representation Learning Workshop*, Montréal, Canada, 2015. URL `http://arxiv.org/abs/1503.02531`.

Julia Hirschberg. Pragmatics and intonation. *The Handbook of Pragmatics*, page 515–537, January 2006. doi: 10.1002/9780470756959.ch23. URL http://dx.doi.org/10.1002/9780470756959.ch23.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*, 2019.

Hsin-Chuan Huang and David E. Eskey. The effects of closed-captioned television on the listening comprehension of intermediate english as a second language (esl) students. *Journal of Educational Technology Systems*, 28(1):75–96, 1999. doi: 10.2190/RG06-LYWB-216Y-R27G. URL https://doi.org/10.2190/RG06-LYWB-216Y-R27G.

Liang Huang, Colin Cherry, Mingbo Ma, Naveen Arivazhagan, and Zhongjun He. Simultaneous translation. In Aline Villavicencio and Benjamin Van Durme, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 34–36, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-tutorials.6. URL https://aclanthology.org/2020.emnlp-tutorials.6.

Wuwei Huang, Mengge Liu, Xiang Li, Yanzhi Tian, Fengyu Yang, Wen Zhang, Jian Luan, Bin Wang, Yuhang Guo, and Jinsong Su. The xiaomi AI lab's speech translation systems for IWSLT 2023 offline task, simultaneous task and speech-to-speech task. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 411–419, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.39. URL https://aclanthology.org/2023.iwslt-1.39.

Muhammad Huzaifah, Kye Min Tan, and Richeng Duan. I2R's end-to-end speech translation system for IWSLT 2023 offline shared task. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 202–210, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.16. URL https://aclanthology.org/2023.iwslt-1.16.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. Multilingual end-to-end speech translation. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 570–577. IEEE, 2019.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311, Online, July 2020. doi: 10.18653/v1/2020.acl-demos.34. URL `https://aclanthology.org/2020.acl-demos.34`.

Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. Non-autoregressive end-to-end speech translation with parallel autoregressive rescoring. *arXiv preprint arXiv:2109.04411*, 2021a.

Hirofumi Inaguma, Brian Yan, Siddharth Dalmia, Pengcheng Guo, Jiatong Shi, Kevin Duh, and Shinji Watanabe. ESPnet-ST IWSLT 2021 offline speech translation system. In Marcello Federico, Alex Waibel, Marta R. Costa-jussà, Jan Niehues, Sebastian Stuker, and Elizabeth Salesky, editors, *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 100–109, Bangkok, Thailand (online), August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwslt-1.10. URL `https://aclanthology.org/2021.iwslt-1.10`.

Sathish Reddy Indurthi, Mohd Abbas Zaidi, Beomseok Lee, Nikhil Kumar Lakumarapu, and Sangha Kim. Language model augmented monotonic attention for simultaneous translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 38–45, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.3. URL `https://aclanthology.org/2022.naacl-main.3`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

Javier Iranzo-Sánchez, Javier Jorge, Pau Baquero-Arnal, Joan Albert Silvestre-Cerdà, Adrià Giménez, Jorge Civera, Albert Sanchis, and Alfons Juan. Streaming cascade-based speech translation leveraged by a direct segmentation model. *Neural Networks*, 142:303–315, 2021.

Javier Iranzo Sanchez, Jorge Civera, and Alfons Juan-Císcar. From simultaneous to streaming machine translation by leveraging streaming history. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6972–6985, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.480. URL https://aclanthology.org/2022.acl-long.480.

Javier Iranzo-Sánchez, Javier Jorge Cano, Alejandro Pérez-González-de Martos, Adrián Giménez Pastor, Gonçal Garcés Díaz-Munío, Pau Baquero-Arnal, Joan Albert Silvestre-Cerdà, Jorge Civera Saiz, Albert Sanchis, and Alfons Juan. MLLP-VRAIN UPV systems for the IWSLT 2022 simultaneous speech translation and speech-to-speech translation tasks. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 255–264, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.22. URL https://aclanthology.org/2022.iwslt-1.22.

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 *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233, 2020. doi: 10.1109/ICASSP40776.2020.9054626.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online, November 2020. doi: 10.18653/v1/2020.findings-emnlp.147. URL https://aclanthology.org/2020.findings-emnlp.147.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184, Brighton, UK, 2019.

Jae-young Jo and Sung-Hyon Myaeng. Roles and utilization of attention heads in transformer-based neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3404–3417, Online, July 2020. doi: 10.18653/v1/2020.acl-main.311. URL https://aclanthology.org/2020.acl-main.311.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. Average Token Delay: A Latency Metric for Simultaneous Translation. In *Proc. INTERSPEECH 2023*, pages 4469–4473, 2023. doi: 10.21437/Interspeech.2023-933.

Alina Karakanta, Matteo Negri, and Marco Turchi. Are subtitling corpora really subtitle-like? In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-It)*, Bari, Italy, November 2019.

Alina Karakanta, Matteo Negri, and Marco Turchi. Is 42 the answer to everything in subtitling-oriented speech translation? In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 209–219, Online, July 2020a. doi: 10.18653/v1/2020.iwslt-1.26. URL `https://www.aclweb.org/anthology/2020.iwslt-1.26`.

Alina Karakanta, Matteo Negri, and Marco Turchi. MuST-cinema: a speech-to-subtitles corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3727–3734, Marseille, France, May 2020b. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.460`.

Alina Karakanta, Matteo Negri, and Marco Turchi. Point Break: Surfing Heterogeneous Data for Subtitle Segmentation. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-It)*, Bologna, Italy, 2020c. URL `http://ceur-ws.org/Vol-2769/paper_78.pdf`.

Alina Karakanta, Marco Gaido, Matteo Negri, and Marco Turchi. Between flexibility and consistency: Joint generation of captions and subtitles. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 215–225, Bangkok, Thailand (online), August 2021a. doi: 10.18653/v1/2021.iwslt-1.26. URL `https://aclanthology.org/2021.iwslt-1.26`.

Alina Karakanta, Sara Papi, Matteo Negri, and Marco Turchi. 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 2021b. URL `https://aclanthology.org/2021.mtsummit-asltrw.4`.

Alina Karakanta, François Buet, Mauro Cettolo, and François Yvon. Evaluating subtitle segmentation for end-to-end generation systems. In *Proceedings of the Thirteenth*

Language Resources and Evaluation Conference, pages 3069–3078, Marseille, France, June 2022. URL https://aclanthology.org/2022.lrec-1.328.

Santosh Kesiraju, Karel Beneš, Maksim Tikhonov, and Jan Černocký. BUT systems for IWSLT 2023 Marathi - Hindi low resource speech translation task. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 227–234, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.19. URL https://aclanthology.org/2023.iwslt-1.19.

Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W Mahoney, and Kurt Keutzer. Squeezeformer: An efficient transformer for automatic speech recognition. *arxiv:2206.00888*, 2022.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, New Orleans, Louisiana, March 2017.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, November 2016. doi: 10.18653/v1/D16-1139. URL https://www.aclweb.org/anthology/D16-1139.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-6503. URL https://aclanthology.org/D19-6503.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio Augmentation for Speech Recognition. In *Proceedings of Interspeech 2015*, pages 3586–3589, Dresden, Germany, September 2015.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online, November 2020. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July 2004. URL https://aclanthology.org/W04-3250.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://aclanthology.org/N03-1017.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-2045.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 115–124, Lisboa, Portugal, November 2020. URL https://aclanthology.org/2020.eamt-1.13.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445.

I. Kožuh and M. Debevc. *Challenges in Social Media Use Among Deaf and Hard of Hearing People*, pages 151–171. Springer International Publishing, Cham, 2018. ISBN 978-3-319-90059-9. doi: 10.1007/978-3-319-90059-9_8. URL https://doi.org/10.1007/978-3-319-90059-9_8.

Helena Kruger. The creation of interlingual subtitles: Semiotics, equivalence and condensation. *Perspectives*, 9(3):177–196, 2001. doi: 10.1080/0907676X.2001.9961416. URL https://doi.org/10.1080/0907676X.2001.9961416.

Jan-Louis Kruger, Natalia Wisniewska, and Sixin Liao. Why subtitle speed matters: Evidence from word skipping and rereading. *Applied Psycholinguistics*, 43(1):211–236, 2022. doi: 10.1017/S0142716421000503.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (Demonstrations)*, pages 66–71, Brussels, Belgium, November 2018. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. CTC-Segmentation of Large Corpora for German End-to-End Speech Recognition. In Alexey Karpov and Rodmonga Potapova, editors, *Speech and Computer*, pages 267–278, Cham, 2020. ISBN 978-3-030-60276-5.

Surafel M. Lakew, Mattia Di Gangi, and Marcello Federico. Controlling the output length of neural machine translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, November 2-3 2019. URL https://aclanthology.org/2019.iwslt-1.31.

Surafel M. Lakew, Marcello Federico, Yue Wang, Cuong Hoang, Yogesh Virkar, Roberto Barra-Chicote, and Robert Enyedi. Machine translation verbosity control for automatic dubbing. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7538–7542, 2021. doi: 10.1109/ICASSP39728.2021.9414411.

Surafel M. Lakew, Yogesh Virkar, Prashant Mathur, and Marcello Federico. Isometric mt: Neural machine translation for automatic dubbing. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6242–6246, 2022. doi: 10.1109/ICASSP43922.2022.9747023.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 245–254, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.27. URL https://aclanthology.org/2022.acl-short.27.

Mathis Lamarre, Catherine Chen, and Fatma Deniz. Attention weights accurately predict language representations in the brain. *bioRxiv*, 2022. doi: 10.1101/2022.12.07.519480. URL https://www.biorxiv.org/content/early/2022/12/07/2022.12.07.519480.

A Lambourne. Subtitle respeaking. *Intralinea, Special Issue on Respeaking*, 2006. URL http://www.intralinea.org/specials/article/Subtitle_respeaking.

Siddique Latif, Aun Zaidi, Heriberto Cuayahuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir. Transformers in speech processing: A survey. *arXiv preprint arXiv:2303.11607*, 2023.

Antoine Laurent, Souhir Gahbiche, Ha Nguyen, Haroun Elleuch, Fethi Bougares, Antoine Thiol, Hugo Riguidel, Salima Mdhaffar, Gaëlle Laperrière, Lucas Maison, Sameer Khurana, and Yannick Estève. ON-TRAC consortium systems for the IWSLT 2023 dialectal and low-resource speech translation tasks. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 219–226, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.18. URL https://aclanthology.org/2023.iwslt-1.18.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533, Barcelona, Spain (Online), December 2020. doi:

10.18653/v1/2020.coling-main.314. URL https://aclanthology.org/2020.coling-main.314.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Lightweight adapter tuning for multilingual speech translation. In *Proc. ACL-IJCNLP 2021*, Online, August 2021. doi: 10.18653/v1/2021.acl-short.103.

Yann LeCun. *Generalization and network design strategies*. 1989.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. Textless speech-to-speech translation on real data. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 860–872, Seattle, United States, July 2022. doi: 10.18653/v1/2022.naacl-main.63. URL https://aclanthology.org/2022.naacl-main.63.

Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Haibo Li, Jing Zheng, Heng Ji, Qi Li, and Wen Wang. Name-aware machine translation. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 604–614, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1059.

Jason Li, Ravi Gadde, Boris Ginsburg, and Vitaly Lavrukhin. Training neural speech recognition systems with synthetic speech augmentation. *arXiv preprint arXiv:1811.00707*, 2018.

Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777, 2014. doi: 10.1109/TASLP.2014.2304637.

Mohan Li and Rama Doddipatla. Non-autoregressive end-to-end approaches for joint automatic speech recognition and spoken language understanding. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 390–397, 2023. doi: 10.1109/SLT54892.2023.10023042.

Xian Li, Changhan Wang, Yun Tang, et al. Multilingual speech translation from efficient finetuning of pretrained models. In *Proc. ACL-IJCNLP 2021*, Online, August 2021. doi: 10.18653/v1/2021.acl-long.68.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. URL `https://aclanthology.org/W04-1013`.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. URL `https://aclanthology.org/L18-1275`.

Dan Liu, Mengge Du, Xiaoxi Li, Yuchen Hu, and Lirong Dai. The USTC-NELSLIP systems for simultaneous speech translation task at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 30–38, Bangkok, Thailand (online), August 2021a. doi: 10.18653/v1/2021.iwslt-1.2. URL `https://aclanthology.org/2021.iwslt-1.2`.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic, November 2021b. URL `https://aclanthology.org/2021.emnlp-main.4`.

Danni Liu, Jan Niehues, and Gerasimos Spanakis. Adapting end-to-end speech recognition for readable subtitles. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 247–256, Online, July 2020a. doi: 10.18653/v1/2020.iwslt-1.30. URL `https://aclanthology.org/2020.iwslt-1.30`.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Proc. Interspeech 2020*, pages 3620–3624, 2020b. doi: 10.21437/Interspeech.2020-2897.

Mengge Liu, Xiang Li, Bao Chen, Yanzhi Tian, Tianwei Lan, Silin Li, Yuhang Guo, Jian Luan, and Bin Wang. BIT-xiaomi's system for AutoSimTrans 2022. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 34–42, Online, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.autosimtrans-1.6. URL `https://aclanthology.org/2022.autosimtrans-1.6`.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-End Speech Translation with Knowledge Distillation. In *Proc. Interspeech 2019*, pages 1128–1132, 2019. doi: 10.21437/Interspeech.2019-2582.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Bridging the modality gap for speech-to-text translation. *arXiv preprint arXiv:2010.14920*, 2020c.

Arle Lommel. Augmented translation: A new approach to combining human and machine capabilities. In Janice Campbell, Alex Yanishevsky, Jennifer Doyon, and Doug Jones, editors, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 5–12, Boston, MA, March 2018. Association for Machine Translation in the Americas. URL `https://aclanthology.org/W18-1905`.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. doi: 10.18653/v1/D15-1166. URL `https://aclanthology.org/D15-1166`.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy, July 2019a. doi: 10.18653/v1/P19-1289. URL `https://aclanthology.org/P19-1289`.

Pingchuan Ma, Stavros Petridis, and Maja Pantic. End-to-end audio-visual speech recognition with conformers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7613–7617. IEEE, 2021a.

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

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 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 2020a. doi: 10.18653/v1/2020.emnlp-demos.19. URL `https://aclanthology.org/2020.emnlp-demos.19`.

Xutai Ma, Juan Pino, and Philipp Koehn. 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 2020b. URL `https://www.aclweb.org/anthology/2020.aacl-main.58`.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7523–7527. IEEE, 2021b.

Dominik Macháček, Ondřej Bojar, and Raj Dabre. MT metrics correlate with human ratings of simultaneous speech translation. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 169–179, Toronto, Canada (in-person and online), July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.12. URL `https://aclanthology.org/2023.iwslt-1.12`.

Dominik Macháček, Raj Dabre, and Ondřej Bojar. Turning whisper into real-time transcription system. *arXiv preprint arXiv:2307.14743*, 2023b.

Matteo Magnini, Giovanni Ciatto, and Andrea Omicini. Knowledge injection of Datalog rules via Neural Network Structuring with KINS. *Journal of Logic and Computation*, page exad037, 06 2023. ISSN 0955-792X. doi: 10.1093/logcom/exad037. URL `https://doi.org/10.1093/logcom/exad037`.

Alison Marsh. Simultaneous Interpreting and Respeaking: a Comparison. Master's thesis, University of Westminster, UK, 2004.

Lara J. Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W Black. Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 303–309, 2015. doi: 10.1109/ASRU.2015.7404809.

Giuseppe Martucci, Mauro Cettolo, Matteo Negri, and Marco Turchi. Lexical Modeling of ASR Errors for Robust Speech Translation. In *Proc. Interspeech 2021*, pages 2282–2286, 2021. doi: 10.21437/Interspeech.2021-265.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA, October 24-25 2005. URL `https://aclanthology.org/2005.iwslt-1.19`.

Evgeny Matusov, Patrick Wilken, and Yota Georgakopoulou. Customizing Neural Machine Translation for Subtitling. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 82–93, Florence, Italy, August 2019. doi: 10.18653/v1/W19-5209. URL `https://www.aclweb.org/anthology/W19-5209`.

Evgeny Matusov, Patrick Wilken, and Christian Herold. Flexible customization of a single neural machine translation system with multi-dimensional metadata inputs. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 204–216, 2020.

Jonathan Mbuya and Antonios Anastasopoulos. GMU systems for the IWSLT 2023 dialect and low-resource speech translation tasks. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 269–276, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.24. URL `https://aclanthology.org/2023.iwslt-1.24`.

Maite Melero, Antoni Oliver, and Toni Badia. Automatic Multilingual Subtitling in the eTITLE Project. In *Proceedings of ASLIB Translating and the Computer 28*, November 2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.6011&rep=rep1`.

Richard J. Harris Michael P. Hinkin and Andrew T. Miranda. Verbal redundancy aids memory for filmed entertainment dialogue. *The Journal of Psychology*, 148(2):161–176, 2014. doi: 10.1080/00223980.2013.767774.

Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, and Tatsuya Kawahara. Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 477–484, 2018. doi: 10.1109/SLT.2018.8639589.

Vikramjit Mitra, Horacio Franco, Richard M. Stern, Julien van Hout, Luciana Ferrer, Martin Graciarena, Wen Wang, Dimitra Vergyri, Abeer Alwan, and John H. L. Hansen. *Robust Features in Deep-Learning-Based Speech Recognition*, pages 187–217. Springer International Publishing, Cham, 2017. ISBN 978-3-319-64680-0. doi: 10.1007/978-3-319-64680-0_8. URL https://doi.org/10.1007/978-3-319-64680-0_8.

Niko Moritz, Takaaki Hori, and Jonathan Le. 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, 2020.

Robert Munro. Crowdsourced translation for emergency response in Haiti: the global collaboration of local knowledge. In *Proceedings of the Workshop on Collaborative Translation: technology, crowdsourcing, and the translator perspective*, Denver, Colorado, USA, October 31 2010. Association for Machine Translation in the Americas. URL https://aclanthology.org/2010.amta-workshop.1.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 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, 2021.

Ha Thanh Nguyen, Trung Kien Vu, Teeradaj Racharak, Le Minh Nguyen, and Satoshi Tojo. Knowledge injection to neural networks with progressive learning strategy. In *Agents and Artificial Intelligence: 12th International Conference, ICAART 2020, Valletta, Malta, February 22–24, 2020, Revised Selected Papers*, page 280–290, Berlin, Heidelberg, 2020a. Springer-Verlag. ISBN 978-3-030-71157-3. doi: 10.1007/978-3-030-71158-0_13. URL https://doi.org/10.1007/978-3-030-71158-0_13.

Thai-Son Nguyen, Sebastian Stueker, Jan Niehues, and Alex Waibel. Improving Sequence-to-sequence Speech Recognition Training with On-the-fly Data Augmentation. In *Proceedings of the 2020 International Conference on Acoustics, Speech, and Signal Processing – IEEE-ICASSP-2020*, Barcelona, Spain, May 2020b.

Jan Niehues, Rolando Cattoni, Sebastian Stüker, Mauro Cettolo, Marco Turchi, and Marcello Federico. The IWSLT 2018 evaluation campaign. In Marco Turchi, Jan Niehues, and Marcello Frederico, editors, *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 2–6, Brussels, October 29-30 2018a. International Conference on Spoken Language Translation. URL `https://aclanthology.org/2018.iwslt-1.1`.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. Low-Latency Neural Speech Translation. In *Proc. Interspeech 2018*, pages 1293–1297, 2018b. doi: 10.21437/Interspeech.2018-1055.

Jan Niehues, Rolando Cattoni, Sebastian Stüker, Matteo Negri, Marco Turchi, Thanh-Le Ha, Elizabeth Salesky, Ramon Sanabria, Loic Barrault, Lucia Specia, and Marcello Federico. The IWSLT 2019 evaluation campaign. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, November 2-3 2019.

Alp Öktem, Mireia Farrús, and Antonio Bonafonte. Prosodic phrase alignment for machine dubbing. *ArXiv*, abs/1908.07226, 2019.

Motoi Omachi, Brian Yan, Siddharth Dalmia, Yuya Fujita, and Shinji Watanabe. Align, write, re-order: Explainable end-to-end speech translation via operation sequence generation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095896.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. doi: 10.18653/v1/W18-6301. URL `https://aclanthology.org/W18-6301`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-4009. URL `https://aclanthology.org/N19-4009`.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Dealing with training and test segmentation mismatch: FBK@IWSLT2021. In *Proceedings of the 18th International Conference on Spoken Language Translation*, pages 84–91, Bangkok, Thailand (online), August 2021a. doi: 10.18653/v1/2021.iwslt-1.8. URL https://aclanthology.org/2021.iwslt-1.8.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Speechformer: Reducing information loss in direct speech translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1698–1706, Online and Punta Cana, Dominican Republic, November 2021b. doi: 10.18653/v1/2021.emnlp-main.127. URL https://aclanthology.org/2021.emnlp-main.127.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online, July 2022a. doi: 10.18653/v1/2022.autosimtrans-1.2. URL https://aclanthology.org/2022.autosimtrans-1.2.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Does simultaneous speech translation need simultaneous models? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 141–153, Abu Dhabi, United Arab Emirates, December 2022b. URL https://aclanthology.org/2022.findings-emnlp.11.

Sara Papi, Alina Karakanta, Matteo Negri, and Marco Turchi. Dodging the data bottleneck: Automatic subtitling with automatically segmented ST corpora. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 480–487, Online only, November 2022c. URL https://aclanthology.org/2022.aacl-short.59.

Sara Papi, Marco Gaido, Alina Karakanta, Mauro Cettolo, Matteo Negri, and Marco Turchi. Direct speech translation for automatic subtitling. *Transactions of the Association for Computational Linguistics*, 11 (in production), 2023a. doi: 10.1162/tacl_a_00607.

Sara Papi, Marco Gaido, and Matteo Negri. Direct models for simultaneous translation and automatic subtitling: FBK@IWSLT2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 159–168, Toronto, Canada (in-person and online), July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.11. URL https://aclanthology.org/2023.iwslt-1.11.

Sara Papi, Marco Gaido, Andrea Pilzer, and Matteo Negri. When good and reproducible results are a giant with feet of clay: The importance of software quality in nlp. *arXiv preprint arXiv:2303.16166*, 2023c.

Sara Papi, Matteo Negri, and Marco Turchi. Attention as a guide for simultaneous speech translation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada, July 2023d. doi: 10.18653/v1/2023.acl-long.745. URL https://aclanthology.org/2023.acl-long.745.

Sara Papi, Marco Turchi, and Matteo Negri. AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation. In *Proc. INTERSPEECH 2023*, pages 3974–3978, 2023e. doi: 10.21437/Interspeech.2023-170.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

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. Interspeech 2019*, pages 2613–2617, 2019. doi: 10.21437/Interspeech.2019-2680. URL http://dx.doi.org/10.21437/Interspeech.2019-2680.

Stephan Peitz, Simon Wiesler, Markus Nußbaum-Thom, and Hermann Ney. Spoken language translation using automatically transcribed text in training. In *Proceedings of the 9th International Workshop on Spoken Language Translation: Papers*, pages 276–283, Hong Kong, Table of contents, December 6-7 2012. URL https://aclanthology.org/2012.iwslt-papers.18.

Elisa Perego. Subtitles and line-breaks: Towards improved readability. *Between Text and Image: Updating research in screen translation*, 78(1):211–223, 2008. doi: 10.1075/btl.78.21per. URL https://doi.org/10.1075/btl.78.21per.

Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002. doi: 10.1162/089120102317341756. URL https://aclanthology.org/J02-1002.

Stelios Piperidis, Iason Demiros, Prokopis Prokopidis, Peter Vanroose, Anja Hoethker, Walter Daelemans, Elsa Sklavounou, Manos Konstantinou, and Yannis Karavidas. Multimodal, multilingual resources in the subtitling process. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/680.pdf.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online), May 2022. doi: 10.18653/v1/2022.iwslt-1.24. URL https://aclanthology.org/2022.iwslt-1.24.

Peter Polák, Danni Liu, Ngoc-Quan Pham, Jan Niehues, Alexander Waibel, and Ondřej Bojar. Towards efficient simultaneous speech translation: CUNI-KIT system for simultaneous track at IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 389–396, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.37. URL https://aclanthology.org/2023.iwslt-1.37.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

Matt Post. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. URL https://www.aclweb.org/anthology/W18-6319.

Tomasz Potapczyk and Pawel Przybysz. SRPOL's system for the IWSLT 2020 end-to-end speech translation task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 89–94, Online, July 2020. doi: 10.18653/v1/2020.iwslt-1.9. URL `https://aclanthology.org/2020.iwslt-1.9`.

Bianca Prandi. Use of CAI tools in interpreters' training: A pilot study. In *Proceedings of Translating and the Computer 37*, London, UK, November 26-27 2015. AsLing. URL `https://aclanthology.org/2015.tc-1.8`.

Bianca Prandi. The use of cai tools in interpreter training: where are we now and where do we go from here? *inTRAlinea*, 01 2020.

Franz Pöchhacker and Aline Remael. New efforts? a competence-oriented task analysis of interlingual live subtitling. *Linguistica Antverpiensia, New Series – Themes in Translation Studies*, 18(0), 2020. ISSN 2295-5739. URL `https://lans-tts.ua.ac.be/index.php/LANS-TTS/article/view/515`.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

Balaji Radhakrishnan, Saurabh Agrawal, Raj Prakash Gohil, Kiran Praveen, Advait Vinay Dhopeshwarkar, and Abhishek Pandey. SRI-B's systems for IWSLT 2023 dialectal and low-resource track: Marathi-Hindi speech translation. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 449–454, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.43. URL `https://aclanthology.org/2023.iwslt-1.43`.

Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2837–2846, 2017.

Matthew Raffel and Lizhong Chen. Implicit memory transformer for computationally efficient simultaneous speech translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12900–12907, Toronto, Canada, July 2023.

Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.816. URL https://aclanthology.org/2023.findings-acl.816.

Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium, November 2018. doi: 10.18653/v1/W18-5431. URL https://aclanthology.org/W18-5431.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 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 2020. doi: 10.18653/v1/2020.acl-main.350. URL https://www.aclweb.org/anthology/2020.acl-main.350.

Katharine Ricke, Laurent Drouet, Ken Caldeira, and Massimo Tavoni. Country-level social cost of carbon. *Nature Climate Change*, 8(10):895, 2018.

Nick Rossenbach, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7069–7073, 2020. doi: 10.1109/ICASSP40776.2020.9053008.

Nicholas Ruiz, Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors. In *Proc. Interspeech 2017*, pages 2635–2639, 2017. doi: 10.21437/Interspeech.2017-1690.

Elizabeth Salesky, Matthias Sperber, and Alexander Waibel. Fluent translations from disfluent speech in end-to-end speech translation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2786–2792, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1285. URL https://aclanthology.org/N19-1285.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.2. URL https://aclanthology.org/2023.iwslt-1.2.

Anoop Sarkar. The challenge of simultaneous speech translation. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Keynote Speeches and Invited Talks*, pages 7–7, Seoul, South Korea, October 2016. URL https://aclanthology.org/Y16-1003.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, nov 2020. ISSN 0001-0782. doi: 10.1145/3381831. URL https://doi.org/10.1145/3381831.

Terrence J. Sejnowski. *The Deep Learning Revolution*. MIT Press, Cambridge, MA, 2018. ISBN 978-0-262-03803-4.

Mark Seligman. Interactive real-time translation via the lnternet. In *Natural Language Processing for the World Wide Web Volume*. AAAI, Oct 1997. URL https://aaai.org/papers/0018-ss97-02-018-interactive-real-time-translation-via-the-lnternet/.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. doi: 10.18653/v1/2020.acl-main.704. URL `https://aclanthology.org/2020.acl-main.704`.

Michael L. Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402, 2013. doi: 10.1109/ICASSP.2013.6639100.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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, August 2016. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL `https://aclanthology.org/2006.amta-papers.25`.

Matthias Sperber and Matthias Paulik. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online, July 2020. doi: 10.18653/v1/2020.acl-main.661. URL `https://www.aclweb.org/anthology/2020.acl-main.661`.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, jan 2014. ISSN 1532-4435.

Sangeeta Srivastava, Yun Wang, Andros Tjandra, Anurag Kumar, Chunxi Liu, Kritika Singh, and Yatharth Saraf. Conformer-based self-supervised learning for non-speech audio tasks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8862–8866, 2022. doi: 10.1109/ICASSP43922.2022.9746490.

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

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

Umut Sulubacak, Ozan Caglayan, Stig-Arne Grönroos, Aku Rouhe, Desmond Elliott, Lucia Specia, and Jörg Tiedemann. Multimodal machine translation through visuals and speech. *Machine Translation*, 34(2):97–147, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 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., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Agnieszka Szarkowska and Olivia Gerber-Morón. Viewers can keep up with fast subtitles: Evidence from eye movements. *PLOS ONE*, 13(6):1–30, 06 2018. doi: 10.1371/journal.pone.0199331. URL https://doi.org/10.1371/journal.pone.0199331.

Agnieszka Szarkowska, Izabela Krejtz, Zuzanna Klyszejko, and Anna Wieczorek. Verbatim, standard, or edited?: Reading patterns of different captioning styles among deaf, hard of hearing, and hearing viewers. *American annals of the deaf*, 156:363–78, 09 2011. doi: 10.1353/aad.2011.0039.

Agnieszka Szarkowska, Izabela Krejtz, Olga Pilipczuk, Łukasz Dutka, and Jan-Louis Kruger. The effects of text editing and subtitle presentation rate on the comprehension and reading patterns of interlingual and intralingual subtitles among deaf, hard of hearing and hearing viewers. *Across Languages and Cultures*, 17(2):183 – 204, 2016. doi: 10.1556/084.2016.17.2.3. URL https://akjournals.com/view/journals/084/17/2/article-p183.xml.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. of 2016 IEEE CVPR*, pages 2818–2826, Las Vegas, Nevada, United States, 2016.

Toshiyuki Takezawa, Tsuyoshi Morimoto, Yoshinori Sagisaka, Nick Campbell, Hitoshi Iida, Fumiaki Sugaya, Akio Yokoo, and Seiichi Yamamoto. A Japanese-to-English speech translation system: ATR-MATRIX. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, November–December 1998.

Derek Tam, Surafel M. Lakew, Yogesh Virkar, Prashant Mathur, and Marcello Federico. Isochrony-Aware Neural Machine Translation for Automatic Dubbing. In *Proc. Interspeech 2022*, pages 1776–1780, 2022. doi: 10.21437/Interspeech.2022-11136.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Brussels, Belgium, October 2018. doi: 10.18653/v1/W18-6304. URL `https://aclanthology.org/W18-6304`.

Yun Tang, Anna Sun, Hirofumi Inaguma, Xinyue Chen, Ning Dong, Xutai Ma, Paden Tomasello, and Juan Pino. Hybrid transducer and attention based encoder-decoder modeling for speech-to-text tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12441–12455, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.695. URL `https://aclanthology.org/2023.acl-long.695`.

Anke Tardel. Effort in Semi-Automatized Subtitling Processes: Speech Recognition and Experience during Transcription. *Journal of Audiovisual Translation*, 3(2):79–102, Dec. 2020. doi: 10.47476/jat.v3i2.2020.131. URL `https://www.jatjournal.org/index.php/jat/article/view/131`.

Jörg Tiedemann. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia, May 30–June 1 2016. URL `https://aclanthology.org/2016.eamt-2.8`.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. Shas: Approaching optimal segmentation for end-to-end speech translation, 2022.

Emiru Tsunoo, Yosuke Kashiwagi, and Shinji Watanabe. Streaming transformer asr with blockwise synchronous beam search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 22–29, 2021. doi: 10.1109/SLT48900.2021.9383517.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy, August 2019. doi: 10.18653/v1/W19-4808. URL `https://aclanthology.org/W19-4808`.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Ricardo Vinuesa, Hossein Azizpour, Iolanda Leite, Madeline Balaam, Virginia Dignum, Sami Domisch, Anna Felländer, Simone Daniela Langhans, Max Tegmark, and Francesco Fuso Nerini. The role of artificial intelligence in achieving the sustainable development goals. *Nature communications*, 11(1):1–10, 2020.

Yogesh Virkar, Marcello Federico, Robert Enyedi, and Roberto Barra-Chicote. Improvements to prosodic alignment for automatic dubbing. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7543–7574, 2021. doi: 10.1109/ICASSP39728.2021.9414966.

Kaisa Vitikainen and Maarit Koponen. Automation in the intralingual subtitling process: Exploring productivity and user experience. *Journal of Audiovisual Translation*, 4(3): 44–65, Dec. 2021. doi: 10.47476/jat.v4i3.2021.197. URL `https://jatjournal.org/index.php/jat/article/view/197`.

Martin Volk, Rico Sennrich, Christian Hardmeier, and Frida Tidström. Machine translation of TV subtitles for large scale production. In *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 53–62, Denver, Colorado, USA, November 4 2010. URL `https://aclanthology.org/2010.jec-1.7`.

Alex Waibel. Speech translation: past, present and future. In *Proc. Interspeech 2004*, pages 353–356, 2004. doi: 10.21437/Interspeech.2004-156.

Alex Waibel, Ajay N. Jain, Arthur E. McNair, Hiroaki Saito, Alexander G. Hauptmann, and Joe Tebelskis. 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 1991.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. Fairseq S2T: Fast speech-to-text modeling with fairseq. 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: System Demonstrations*, pages 33–39, Suzhou, China, December 2020a. URL `https://aclanthology.org/2020.aacl-demo.6`.

Changhan Wang, Anne Wu, and Juan Pino. Covost 2: A massively multilingual speech-to-text translation corpus, 2020b.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. 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 993–1003, Online, August 2021. doi: 10.18653/v1/2021.acl-long.80. URL `https://aclanthology.org/2021.acl-long.80`.

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

Minghan Wang, Jiaxin Guo, Yinglu Li, Xiaosong Qiao, Yuxia Wang, Zongyao Li, Chang Su, Yimeng Chen, Min Zhang, Shimin Tao, Hao Yang, and Ying Qin. The HW-TSC's simultaneous speech translation system for IWSLT 2022 evaluation. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 247–254, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.21. URL `https://aclanthology.org/2022.iwslt-1.21`.

Peidong Wang, Eric Sun, Jian Xue, Yu Wu, Long Zhou, Yashesh Gaur, Shujie Liu, and Jinyu Li. LAMASSU: A Streaming Language-Agnostic Multilingual Speech Recognition and Translation Model Using Neural Transducers. In *Proc. INTERSPEECH 2023*, pages 57–61, 2023a. doi: 10.21437/Interspeech.2023-2004.

Zhipeng Wang, Yuhang Guo, and Shuoying Chen. BIT's system for multilingual track. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 455–460, Toronto, Canada (in-person and online), July 2023b. doi: 10.18653/v1/2023.iwslt-1.44. URL `https://aclanthology.org/2023.iwslt-1.44`.

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

Orion Weller, Matthias Sperber, Christian Gollan, and Joris Kluivers. Streaming models for joint speech recognition and translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2533–2539, Online, April 2021. doi: 10.18653/v1/2021.eacl-main.216.

Patrick Wilken, Panayota Georgakopoulou, and Evgeny Matusov. SubER - a metric for automatic evaluation of subtitle quality. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 1–10, Dublin, Ireland (in-person and online), May 2022. URL `https://aclanthology.org/2022.iwslt-1.1`.

Zhihang Xie. The BIGAI offline speech translation systems for IWSLT 2023 evaluation. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 123–129, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.7. URL `https://aclanthology.org/2023.iwslt-1.7`.

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

Chen Xu, Xiaoqian Liu, Xiaowen Liu, Tiger Wang, Canan Huang, Tong Xiao, and Jingbo Zhu. The NiuTrans end-to-end speech translation system for IWSLT 2021

offline task. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 92–99, Bangkok, Thailand (online), August 2021. doi: 10.18653/v1/2021.iwslt-1.9. URL https://aclanthology.org/2021.iwslt-1.9.

Chen Xu, Rong Ye, Qianqian Dong, Chengqi Zhao, Tom Ko, Mingxuan Wang, Tong Xiao, and Jingbo Zhu. Recent advances in direct speech-to-text translation. *arXiv preprint arXiv:2306.11646*, 2023.

Jitao Xu, François Buet, Josep Crego, Elise Bertin-Lemée, and François Yvon. Joint generation of captions and subtitles with dual decoding. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 74–82, Dublin, Ireland (in-person and online), May 2022. doi: 10.18653/v1/2022.iwslt-1.7. URL https://aclanthology.org/2022.iwslt-1.7.

Jian Xue, Peidong Wang, Jinyu Li, Matt Post, and Yashesh Gaur. Large-Scale Streaming End-to-End Speech Translation with Neural Transducers. In *Proc. Interspeech 2022*, pages 3263–3267, 2022. doi: 10.21437/Interspeech.2022-10953.

Sane Yagi. Studying style in simultaneous interpretation. *Meta*, 45(3):520–547, 2000. doi: https://doi.org/10.7202/004626ar.

Brian Yan, Jiatong Shi, Soumi Maiti, William Chen, Xinjian Li, Yifan Peng, Siddhant Arora, and Shinji Watanabe. CMU's IWSLT 2023 simultaneous speech translation system. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 235–240, Toronto, Canada (in-person and online), July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.iwslt-1.20. URL https://aclanthology.org/2023.iwslt-1.20.

Rong Ye, Mingxuan Wang, and Lei Li. End-to-End Speech Translation via Cross-Modal Progressive Training. In *Proc. Interspeech 2021*, 2021. doi: 10.21437/Interspeech.2021-1065.

Rong Ye, Mingxuan Wang, and Lei Li. Cross-modal contrastive learning for speech translation. In *Proc. NAACL 2022*, Seattle, United States, July 2022. doi: 10.18653/v1/2022.naacl-main.376.

Ching-Feng Yeh, Jay Mahadeokar, Kaustubh Kalgaonkar, Yongqiang Wang, Duc Le, Mahaveer Jain, Kjell Schubert, Christian Fuegen, and Michael L Seltzer. Transformer-transducer: End-to-end speech recognition with self-attention. *arXiv preprint arXiv:1910.12977*, 2019.

Mohd Abbas Zaidi, Beomseok Lee, Nikhil Kumar Lakumarapu, Sangha Kim, and Chanwoo Kim. Decision attentive regularization to improve simultaneous speech translation systems. *arXiv preprint arXiv:2110.15729*, 2021.

Mohd Abbas Zaidi, Beomseok Lee, Sangha Kim, and Chanwoo Kim. Cross-Modal Decision Regularization for Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 116–120, 2022. doi: 10.21437/Interspeech.2022-10617.

Xingshan Zeng, Liangyou Li, and Qun Liu. 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 2021. doi: 10.18653/v1/2021.findings-acl.218. URL `https://aclanthology.org/2021.findings-acl.218`.

Xingshan Zeng, Pengfei Li, Liangyou Li, and Qun Liu. End-to-end simultaneous speech translation with pretraining and distillation: Huawei Noah's system for AutoSimTranS 2022. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 25–33, Online, July 2022. doi: 10.18653/v1/2022.autosimtrans-1.5. URL `https://aclanthology.org/2022.autosimtrans-1.5`.

Thomas Zenkel, Joern Wuebker, and John DeNero. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*, 2019.

Richard Zens, Franz Josef Och, and Hermann Ney. Phrase-based statistical machine translation. In Matthias Jarke, Gerhard Lakemeyer, and Jana Koehler, editors, *KI 2002: Advances in Artificial Intelligence*, pages 18–32, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45751-0.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium, October-November 2018. doi: 10.18653/v1/D18-1049. URL `https://aclanthology.org/D18-1049`.

Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. Learning adaptive segmentation policy for end-to-end simultaneous translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7862–7874, Dublin, Ireland, May 2022a. Associa-

tion for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.542. URL https://aclanthology.org/2022.acl-long.542.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Haifeng Wang, Liang Huang, Qun Liu, Julia Ive, and Wolfgang Macherey. Findings of the third workshop on automatic simultaneous translation. In Julia Ive and Ruiqing Zhang, editors, *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 1–11, Online, July 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.autosimtrans-1.1. URL https://aclanthology.org/2022.autosimtrans-1.1.

Shaolei Zhang and Yang Feng. Information-transport-based policy for simultaneous translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 992–1013, Abu Dhabi, United Arab Emirates, December 2022. URL https://aclanthology.org/2022.emnlp-main.65.

Shaolei Zhang and Yang Feng. End-to-end simultaneous speech translation with differentiable segmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7659–7680, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.485. URL https://aclanthology.org/2023.findings-acl.485.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online, July 2020. doi: 10.18653/v1/2020.acl-main.254. URL https://aclanthology.org/2020.acl-main.254.

Xinyuan Zhou, Jianwei Cui, Zhongyi Ye, Yichi Wang, Luzhen Xu, Hanyi Zhang, Weitai Zhang, and Lirong Dai. Submission of USTC's system for the IWSLT 2023 - offline speech translation track. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 194–201, Toronto, Canada (in-person and online), July 2023. doi: 10.18653/v1/2023.iwslt-1.15. URL https://aclanthology.org/2023.iwslt-1.15.

Qinpei Zhu, Renshou Wu, Guangfeng Liu, Xinyu Zhu, Xingyu Chen, Yang Zhou, Qingliang Miao, Rui Wang, and Kai Yu. The AISP-SJTU simultaneous translation system for IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 208–215, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.16. URL `https://aclanthology.org/2022.iwslt-1.16`.