UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Direct Speech Translation

Toward High-Quality, Inclusive, and Augmented Systems

Marco Gaido

Advisors

Marco Turchi	Zoom Video Communications
Matteo Negri	Fondazione Bruno Kessler

Committee

Marta R. Costa-jussà	Meta AI
Jan Niehues	Karlsruhe Institute of Technology
Laurent Besacier	Naver Labs

April 2023

Abstract

When this PhD started, the translation of speech into text in a different language was mainly tackled with a cascade of automatic speech recognition (ASR) and machine translation (MT) models, as the emerging direct speech translation (ST) models were not yet competitive. To close this gap, part of the PhD has been devoted to improving the quality of direct models, both in the simplified condition of test sets where the audio is split into well-formed sentences, and in the realistic condition in which the audio is automatically segmented. First, we investigated how to transfer knowledge from MT models trained on large corpora. Then, we defined encoder architectures that give different weights to the vectors in the input sequence, reflecting the variability of the amount of information over time in speech. Finally, we reduced the adverse effects caused by the suboptimal automatic audio segmentation in two ways: on one side, we created models robust to this condition; on the other, we enhanced the audio segmentation itself. The good results achieved in terms of overall translation quality allowed us to investigate specific behaviors of direct ST systems, which are crucial to satisfy real users' needs. On one side, driven by the ethical goal of inclusive systems, we disclosed that established technical choices geared toward high general performance (statistical word segmentation of the target text, knowledge distillation from MT) cause an exacerbation of the gender representational disparities in the training data. Along this line of work, we proposed mitigation techniques that reduce the gender bias of ST models, and showed how gender-specific systems can be used to control the translation of gendered words related to the speakers, regardless of their vocal traits. On the other side, motivated by the practical needs of interpreters and translators, we evaluated the potential of direct ST systems in the "augmented translation" scenario, focusing on the translation and recognition of named entities (NEs). Along this line of work, we proposed solutions to cope with the major weakness of ST models (handling person names), and introduced direct models that jointly perform ST and NE recognition showing their superiority over a pipeline of dedicated tools for the two tasks. Overall, we believe that this thesis moves a step forward toward adopting direct ST systems in real applications, increasing the awareness of their strengths and weaknesses compared to the traditional cascade paradigm.

Keywords

[direct speech translation, audio segmentation, gender bias, named entities, augmented translation]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Context

We live in a globalized world, where audiovisual content is the most widespread mean of communication. Every day millennials spend more than 17 (overlapping) hours consuming audiovisual content,[1] and American people listen to roughly 1 hour and a half of audio content.[2] As such, the access to this huge amount of online material covers paramount importance and, foreseeing its value, UNESCO promoted multilingualism to ensure universal access since 2003.[3] However, the translation of all the speech content into any language represents a pipe dream that cannot be addressed only through the human workforce of professionals, advocating for automatic systems to reduce the burden of translators. From these considerations, it is no surprise that the automatic translation of speech is gaining increasing interest for its potential pervasiveness in our daily life, with applications that range from subtitling (Matusov et al., 2019), travel conversations (Takezawa et al., 1998), and lecture translation (Fügen, 2009)

[1] https://www.entrepreneur.com/growing-a-business/millennials-spend-18-hours-a-day-consuming-media-and/232062

[2] https://www.insiderintelligence.com/chart/243762/digital-audio-average-time-spent-us-2018-2022-minutes-per-day-among-population-change

[3] https://www.unesco.org/en/communication-information/multilingualism-cyberspace

to documentation of endangered languages and crisis response (Bansal et al., 2017), or even humanitarian expeditions (Black et al., 2002).

Although researchers have been confronted with the problem since the late 80s (Stentiford and Steer, 1988; Waibel et al., 1991), the translation of a speech segment (or utterance) into its textual content in a different language is still a challenging task to perform automatically. The input, in fact, has to be converted along two dimensions: the modality (from speech to text), and the language. Due to the complexity of the task, the problem has been decomposed into simpler parts, among which two are fundamental: first, an automatic speech recognition (ASR) system covers the modality transformation (from speech to text); then, a machine translation (MT) system translates the produced text into the target language. This solution, known as *cascade*, has been the standard approach for decades, and the deep learning revolution (LeCun et al., 2015; Sejnowski, 2018) has initially involved only its individual components, without changing the overall composition and the interaction among the constituent systems.

In this revolution, phrase-based statistical MT (SMT) models (Zens et al., 2002; Koehn et al., 2003) were replaced by neural MT (NMT) networks (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014). In ASR, the traditional approach relied on Gaussian mixture (Juang et al., 1986) models to estimate the acoustic probability of a word in a given frame or set of frames, and on Hidden Markov Models (HMM) to explore the sequence of words considered acceptable by a language model (LM) (Lamere et al., 2003; Schiel, 1999). Similarly to MT, first, the acoustic models have switched from Gaussian mixture (Juang et al., 1986) to deep neural networks (DNN) (Hinton et al., 2012); then, the whole task has been performed with a single end-to-end DNN (Graves and Jaitly, 2014; Chorowski et al., 2014).

This PhD journey started in 2019, a few years after the introduction of the first end-to-end, or *direct*, speech-to-text translation (ST) models (Bérard et al., 2016; Weiss et al., 2017). Such models do not leverage intermediate representations to perform the task (like the transcripts of cascade solutions) and all their parameters are jointly trained toward the ST task. Back then (in 2019), these models were struggling in achieving the same translation quality as their cascade counterpart (Niehues et al., 2019), although the gap dramatically decreased from the previous year (Niehues et al., 2018). As such, closing this gap was the main focus of research efforts (Di Gangi et al. 2019c; Bahar et al. 2019a among others).

This thesis not only joins the endeavors to increase the quality of direct models, but – in light of the considerable reduction of the performance gap with cascade systems – explores ancillary challenges to assess their specific strengths and weaknesses in different use cases, and scenarios.

1.2 The Challenges

Translation quality is essential for an automatic ST system. However, many other factors – such as efficiency, robustness, flexibility, customizability – are critical in determining the success of an approach, as many applications pose specific constraints. For instance, simultaneous translation requires low latency[4] (which implies model efficiency), while subtitling/dubbing calls for short translations to comply with space and reading speed constraints (Diaz-Cintas and Remael, 2007). In addition, the presence of non-native speakers or noisy conditions increase the complexity of the task (Ansari et al., 2020) and the recent awareness of the need for inclusive technologies, representative of all groups and individuals, constitutes a crucial theme for every application (Hovy and Spruit, 2016; Blodgett et al., 2020). At last,

[4] 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).

the importance of user perspective and the centrality of the human are leading to new paradigms for translation systems (Lommel, 2018), in which the key is the information provided to the user (professional translators, post-editors, or end users), rather than the simple fluency and overall correctness of the translation.

In light of these considerations, and with the goal of exploring the strengths and weaknesses of direct ST models in future production environments, this PhD was directed toward two objectives: *i)* closing the overall translation quality gap with cascade systems, and *ii)* investigating aspects that are neglected by the coarse-grained indication of the holistic quality (Callison-Burch et al., 2006) provided by automatic metrics, such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006), or even neural ones like COMET (Rei et al., 2020), which are relatively insensitive to errors on gender inflections (Bentivogli et al., 2020) and named entities (Amrhein and Sennrich, 2022). With respect to objective *i)*, on one side we worked on training procedures and architectural solutions aimed at improving the translation quality of direct ST systems with reasonable computational costs (§1.2.1). On the other, we focused on how to limit the quality drops observed when the audio is not segmented according to a known reference but has to be automatically segmented into chunks processable by ST models (§1.2.2). Regarding objective *ii)*, we first devoted ourselves to increasing the inclusivity of direct ST models with respect to underrepresented gender categories (§1.2.3). Then, we studied how to integrate direct ST systems in the "augmented translation" paradigm, which requires enhancing and enriching the output with additional information aimed at helping users' comprehension (§1.2.4).

1.2.1 Direct ST Quality and Efficiency

The introduction of the *direct* approach is motivated by its theoretical and practical advantages (Sperber and Paulik, 2020): *i)* during the translation phase it has access to information present in the audio that is lost in the transcripts (e.g. prosody), *ii)* there is no *error propagation* (Ruiz and Federico, 2014) (in cascade systems the errors introduced by the ASR are propagated to the MT, which has no cues to recover them), *iii)* the latency is lower (as data flows through a single system instead of two), and *iv)* the management is easier (as there is a single model to maintain and no integration between separate modules is needed).

On the downside, direct ST suffers from *i)* the lack of large ST training corpora, *ii)* the complexity of addressing the task with a single model, and *iii)* problems related to managing long input sequences. Regarding *i)*, the biggest ST corpus currently available is MuST-C (Di Gangi et al., 2019a), which contains ∼500 hours of recorded TED talks resulting in ∼250K triplets *(audio, transcript, translation)*. In comparison, many ASR corpora are available and contain twice the number of hours (Panayotov et al., 2015), while MT corpora often have more than 50M sentence pairs (Tiedemann, 2016), two orders of magnitude more than ST data. The issues *ii)* and *iii)*, instead, lead to a challenging trade-off between the size of an ST model and its computational cost. Indeed, the input of the network is a sequence of samples collected from the audio with a high frequency, typically one sample every 10ms. The resulting sequence length is usually one order of magnitude higher than the corresponding MT input sequence derived from the text. For this reason, dedicated architectures are needed to avoid prohibitive computational and memory footprint, especially in modern architectures based on the *self-attention* mechanism (Vaswani et al., 2017) that has a quadratic complexity with respect to the length of the input sequence.

In this thesis, we address these limitations in §3, with a particular focus on computational efficiency, in line with the rising concerns about the social and environmental impact of expensive practices (Strubell et al., 2019).

1.2.2 Audio Segmentation

Direct and cascade ST systems are usually tested on benchmarks in which the audio has been segmented into short segments of speech corresponding to well-formed sentences (Di Gangi et al., 2019a; Iranzo-Sánchez et al., 2020). Indeed, these test sets split continuous speech into utterances according to strong punctuation marks in the transcripts (which are known in advance), reflecting linguistic criteria related to sentence well-formedness. This (manual) segmentation is optimal, as it allows ST systems to potentially generate correct outputs even for languages with different syntax and word order (e.g. subject-verb-object vs subject-object-verb). However, it does not represent a realistic condition, as production deployments expose ST systems to long, unsegmented audio streams, whose content is totally unknown. In this scenario, the traditional approach consists in splitting the audio on speaker silences – considered as a *proxy* of clause boundaries – with a Voice Activity Detection (VAD) tool (Sohn et al., 1999). Since the produced segmentation is not driven by syntactic information (unlike that of the training corpora), final performance on downstream tasks is exposed to considerable degradation (Sinclair et al., 2014).

In cascade systems, the impact of a syntax-unaware segmentation can be mitigated by means of dedicated components that re-segment the ASR transcripts, so to feed the MT model with well-formed sentences (Matusov et al., 2006). The absence of intermediate transcripts makes this solution unfeasible for direct systems, whose performance is therefore highly sensitive to sub-optimal audio segmentation. The thesis covers our work on this problem in §4 with a two-fold approach: on one side we improve the way the audio is segmented; on the other, we build models that are more robust to the mismatch between the well-formed training data and suboptimal splits supplied at inference time.

1.2.3 Gender Bias

The term "bias" comes from cognitive sciences (Tversky and Kahneman, 1973, 1974) and is conceived as the divergence from an expected value (Glymour and Herington, 2019; Shah et al., 2020). As such, by gender bias in automatic translation systems we refer to the overproduction of masculine references in their outputs (Cho et al., 2019; Bentivogli et al., 2020), and to feminine/masculine associations perpetuating traditional gender roles and stereotypes (Prates et al., 2020; Stanovsky et al., 2019). This attested systemic bias can directly affect the users of such technology by diminishing their gender identity or further exacerbating existing social inequalities and access to opportunities for women (Barocas et al., 2017; Crawford, 2017).

The problem is exacerbated whenever systems are required to overtly and formally express the speaker's gender in the target languages while translating from languages that do not convey such information. Indeed, languages with grammatical gender, such as French and Italian, display a complex morphosyntactic and semantic system of gender agreement (Hockett, 1958; Corbett, 1991) relying on feminine/masculine markings reflecting speakers' gender on numerous parts of speech whenever they are talking about themselves (e.g., En: *I've never **been** there* – It: *Non ci sono mai **stata/stato***). Differently, English is a natural gender language (Hellinger and Bußman, 2001) that mostly conveys gender via its pronoun system, but only for third-person pronouns (*he/she*), thus to refer to an entity other than the speaker. In light of the importance and the scale of the problem and as a first step towards the design of more inclusive technology, in §5 we explore mitigation strategies designed to reduce the gender bias in direct ST systems in the challenging condition of target languages with grammatical gender.

1.2.4 Augmented Speech Translation

"Augmented translation" (Lommel, 2018) is an emerging approach in automatic translation that aims at tightly integrating translation systems with humans (either professional translators and post-editors or end users). Drawing inspiration from augmented reality, where real-world vision is complemented with relevant information, in augmented translation the output is enriched and linked with information about useful concepts and named entities (NEs) to help users' understanding.[5] On one side, this can ease, speed up, and improve the generation of fluent and high-quality translations by professional translators and post-editors; on the other, it provides end users with additional information that may be needed to fully understand a sentence, especially in highly specialized domains. Augmented ST hence requires not only an accurate translation of the source speech, but paramount importance is given to the accurate rendering of the NEs involved, and to their recognition in the output text as well as, possibly, their linking to external knowledge bases.

Motivated by the practical relevance of the problem, in §6 the ability of direct ST models to properly handle NEs is first assessed, compared to that of cascade systems, and then improved, together with their capability to recognize which words belong to NE and which do not.

[5] `https://intelligent-information.blog/en/augmented-translation-puts-translators-back-in-the-center/`

1.3 Contributions

Following the partition into four pillars outlined in the previous section, the main contributions of this thesis are listed below.[6]

1.3.1 Direct ST Quality and Efficiency

- The study of solutions to transfer the knowledge learned by MT models trained on large amounts of parallel textual data into direct ST models.
  - Gaido, M., Di Gangi, M. A., Negri, M., Turchi, M. (2020). End-to-End Speech-Translation with Knowledge Distillation: FBK@ IWSLT2020. In Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020 (pp. 80-88).
  - Gaido, M., Di Gangi, M. A., Negri, M., Turchi, M. (2020). On Knowledge Distillation for Direct Speech Translation. In Seventh Italian Conference on Computational Linguistics, CLiC-it 2020 (Vol. 2769). **Best Paper Award.**
  - Gaido, M., Negri, M., Turchi, M. (2022). Direct Speech-to-Text Translation Models as Students of Text-to-Text Models. Italian Journal of Computational Linguistics, IJCoL 8(8-1).
- The proposal of architectural solutions to compress the input audio sequence, limiting the information loss, with the twofold goal of improving the translation quality and reducing the computational cost.
  - Gaido, M., Cettolo, M., Negri, M., Turchi, M. (2021). CTC-based Compression for Direct Speech Translation. In Proceedings of the

[6] All the referenced papers are the results of a collaborative effort with the co-authors. For most of the works, I am the first author, meaning that I was responsible for the research ideas, the implementations, the experiments, the evaluation, the result discussion, and the writing, with the support and feedback of the other authors. For the works marked with *, I am first co-author with equal contributions. For all works where I am not the first author, a description of my contributions is provided.

16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL (pp. 690-696).

- Papi, S.[*], Gaido, M.[*], Negri, M., Turchi, M. (2021). Speechformer: Reducing Information Loss in Direct Speech Translation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021 (pp. 1698-1706).

  My contributions: equal contribution to research ideas, implementation, experiments, and writing.

- Gaido, M.[*], Papi, S.[*], Fucci, D., Fiameni, G., Negri, M., Turchi, M. (2022). Efficient yet Competitive Speech Translation: FBK@ IWSLT2022. In Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT 2022 (pp. 177-189).

  My contributions: the work on offline speech translation systems.

1.3.2 Audio Segmentation

- The proposal of methods to build direct ST models that are robust to imperfect automatic audio segmentation.

  - Gaido, M., Di Gangi, M. A., Negri, M., Cettolo, M., Turchi, M. (2020). Contextualized Translation of Automatically Segmented Speech. Proceedings of Interspeech 2020, 1471-1475.

  - Papi, S., Gaido, M., Negri, M., Turchi, M. (2021). Dealing with training and test segmentation mismatch: FBK@ IWSLT2021. In Proceedings of the 18th International Conference on Spoken Language Translation, IWSLT 2021 (pp. 84-91).

    My contributions: system design, guidance on implementation and experiments.

- The proposal of automatic audio segmentation approaches that limit the translation quality degradation with respect to manual audio segmentation and are applicable efficiently to audio streams.

  - Gaido, M., Negri, M., Cettolo, M., Turchi, M. (2021). Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation. In Proceedings of The Fourth International Conference on Natural Language and Speech Processing, ICNLSP 2021 (pp. 55-62).

1.3.3 Gender Bias

- A survey on gender bias in the related field of MT, where my particular focus was directed on the technical solutions proposed as mitigation strategies.

  - Savoldi, B., Gaido, M., Bentivogli, L., Negri, M., Turchi, M. (2021). Gender Bias in Machine Translation. Transactions of the Association for Computational Linguistics, TACL 9, 845-874.

    My contributions: literature review of the technical solutions and contribution to writing.

- The study of different solutions to integrate and control the information of the speaker's preferred gender in direct ST systems.

  - Gaido, M.*, Savoldi, B.*, Bentivogli, L., Negri, M., Turchi, M. (2020). Breeding Gender-aware Direct Speech Translation Systems. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020 (pp. 3951-3964). **Outstanding Paper.**

    My contributions: implementation, experiments, and automatic evaluation.

- An analysis of the effect of different word segmentation methods on gender bias in direct ST with the definition of a mitigation strategy that takes account of both translation quality and gender translation, and the creation of a benchmark extending MuST-SHE (Bentivogli et al., 2020) – a test set made of TED talks focused on gender evaluation – for fine-grained evaluation of morphosyntactic capabilities of ST systems.

- Gaido, M.*, Savoldi, B.*, Bentivogli, L., Negri, M., Turchi, M. (2021). How to Split: the Effect of Word Segmentation on Gender Bias in Speech Translation. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 3576-3589).

  My contributions: research idea, implementation, experiments, and automatic evaluation.

- Savoldi, B., Gaido, M., Bentivogli, L., Negri, M., Turchi, M. (2022). Under the Morphosyntactic Lens: A Multifaceted Evaluation of Gender Bias in Speech Translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022 (pp. 1807-1824).

  My contributions: implementation, experiments, and automatic evaluation of the systems.

- Savoldi, B., Gaido, M., Bentivogli, L., Negri, M., Turchi, M. (2022). On the Dynamics of Gender Learning in Speech Translation. In Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing, GeBNLP 2022 (pp. 94-111).

  My contributions: implementation, experiments, and automatic evaluation.

1.3.4 Augmented Speech Translation

- A systematic analysis of the behavior of state-of-the-art ST systems in translating NEs and terminology, with the release of a novel benchmark built from European Parliament speeches annotated with NEs and terminology.

  - Gaido, M., Rodríguez, S., Negri, M., Bentivogli, L., Turchi, M. (2021). 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, EMNLP 2021 (pp. 1707-1716).

- The proposal of different methods to improve the quality of the translation of NEs, in particular of person names, both with and without exploiting contextual information.

  - Gaido, M., Negri, M., Turchi, M. (2022). Who Are We Talking About? Handling Person Names in Speech Translation. In Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT 2022 (pp. 62-73). **Best Paper Award.**
  - Gaido, M., Tang, Y., Kulikov, I., Huang, R., Gong, H., Inaguma, H. (2022). Named Entity Detection and Injection for Direct Speech Translation. Accepted at ICASSP 2023. © 2023 IEEE.

- The introduction of direct models that jointly perform ST and NEs, with an extensive comparison among them accounting for both quality and efficiency.

  - Gaido, M., Papi, S., Negri, M., Turchi, M. (2022). Joint Speech Translation and Named Entity Recognition. Under review at Interspeech 2023.

1.3.5 Other contributions

Open Source Codebase. On account of my previous working experience in software engineering and of my deep belief in open source developed working in the Apache community, I would like to highlight an initiative I advocated since my first day as a PhD student, which was supported by my advisors and the other people of the MT unit at FBK: the release of the code for most of the works presented in this thesis (and for the other recent works of the group) with MIT license in an open-source codebase[7] derived from a fork of the fairseq (Ott et al., 2019) repository. The repository is built following software engineering practices to enforce the quality of the software: *i)* unit tests (UTs) are added and executed for every new commit with standard continuous integration (CI) pipeline enforcing that new contributions do not break existing features and cause regressions; *ii)* templates for merge requests (MRs) are required to provide context and documentation regarding the code added, and these descriptions are included in commit messages; *iii)* all contributions are internally peer-reviewed by another member of the group to improve the readability of the code and style coherence of the codebase. The adoption of these practices is a guarantee of the functionality, code quality, and stability of the codebase, both for the MT unit at FBK and for people interested in using and/or contributing to our repository, and distinguish it from the original fairseq repository where breaking changes are frequently introduced without notification and documentation, due to the lack of working and complete UTs and CI. Unfortunately, this initiative started in 2021, and not all the code of previous works – developed for an older fairseq version – has been migrated. For this reason, the code for papers from 2020 and part of 2021 can be found in a previous repository.[8]

Other Works. Lastly, I have been involved in collaboration with other PhD students and researches within the MT unit that contributed to my personal, professional, and scientific growth. These works led to publications in which my contribution was not prevalent and/or on topics related with

[7] `https://github.com/hlt-mt/fbk-fairseq`

[8] `https://github.com/mgaido91/FBK-fairseq-ST`

this thesis, but not strictly part of it. As such, they have not been included in the discussion, but, to acknowledge their importance as part of my PhD experience, I list them here:

– Di Gangi, M. A., Gaido, M., Negri, M., Turchi, M. (2020). On Target Segmentation for Direct Speech Translation. In Proceedings of the 14th Conference of the Association for Machine Translation in the Americas, AMTA 2020 (Volume 1: Research Track) (pp. 137-150).

– Karakanta, A., Gaido, M., Negri, M., Turchi, M. (2021). Between Flexibility and Consistency: Joint Generation of Captions and Subtitles. In Proceedings of the 18th International Conference on Spoken Language Translation, IWSLT 2021 (pp. 215-225).

– Bentivogli, L., Cettolo, M., Gaido, M., Karakanta, A., Martinelli, A., Negri, M., Turchi, M. (2021). Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference?. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), ACL 2021 (pp. 2873-2887).

– Bentivogli, L., Cettolo, M., Gaido, M., Karakanta, A., Negri, M., Turchi, M. (2022). Extending the MuST-C Corpus for a Comparative Evaluation of Speech Translation Technology. In Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, EAMT 2022 (pp. 359-360).

– 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, AutoSimTrans 2022 (pp. 12-17).

– Papi, S., Gaido, M., Negri, M., Turchi, M. (2022). Does Simultaneous Speech Translation need Simultaneous Models?. In Findings of the Association for Computational Linguistics: EMNLP 2022.

1.4 Structure of the Thesis

The structure of the thesis reflects the challenges and contributions listed in the previous sections. Specifically, after a background chapter (§2) involving the fundamental concepts useful for the understanding of the thesis content, the thesis continues with a chapter for each of the four pillars (direct ST quality and efficiency in §3, audio segmentation in §4, gender bias in §5, and augmented ST in §6), describing the corresponding contributions. The conclusions chapter (§7) closes the thesis with remarks from the work carried out in these three years, the limitations, and future research directions.

Chapter 2

Background

The chapter contains a brief description of the technical concepts useful to understand the content of the thesis. Complementary to the content of this chapter, each of the four following chapters (§3, §4, §5, §6) has dedicated realted works with the details specific to the topic.

As it would be impossible to include all the required technical concepts in the thesis, a preliminary knowledge of the basics of neural networks and deep learning is assumed. The reader can refer to the literature thoroughly describing the topics they are not familiar with: for instance, (Goodfellow et al., 2016) is a detailed resource meant for beginners that covers all basic theory – including the mathematical and statistical theory – and more advanced topics.

2.1 Deep Learning

Deep learning is the field of the artificial intelligence (AI) that studies neural networks, which "are composed of multiple processing layers to learn representations of data with multiple levels of abstraction" (LeCun et al., 2015). The multiple layers are made of a set of parameters (usually referred as $\boldsymbol{\theta}$ in literature) that have to be "learned", i.e. optimized for a specific task. The task is defined by means of an *objective function* (or *criterion*) that can be either maximized or minimized. In the second case – the most common in deep learning[1] – they take the name of *loss function* (Wald, 1949).

In the traditional scenario of supervised learning, loss functions take as input the probability distribution generated by the neural network and a *reference* probability distribution, defined by a training set, which is considered the ground truth. In particular, the most widespread loss function is the *cross entropy*, which can be seen as a particular case of the KL-divergence (Kullback and Leibler, 1951) that is a measure of the distance between the two probability distributions. The KL-divergence is formally defined as:

$$KL(p||q) = \sum p(x) * log \frac{p(x)}{q(x)} = \sum_{x \in X} \Big(p(x) * logp(x) - p(x) * logq(x) \Big) \quad (2.1)$$

which measures the *closeness* of q to p, i.e. how much information is lost when using q to approximate p. As, in the considered case of supervised learning, we are interested in how likely is an output y among all the possible Y outputs given an input x that is part of the training set X, the above equation can be rewritten as:

$$\begin{aligned} L(X) &= \sum_{x \in X} \sum_{y \in Y} p(y|x) * \frac{p(y|x)}{q(y|x)} \\ &= \sum_{x \in X} \sum_{y \in Y} p(y|x) * log(p(y|x)) - \sum_{x \in X} \sum_{y \in Y} p(y|x) * log(q(y|x)) \end{aligned} \quad (2.2)$$

where q is the probability distribution generated by the model and p is the reference distribution. Since the first term does not depend on the model probability, we can omit it in the optimization. In addition, we can

[1] Functions that should be maximized are commonly negated to obtain a function that has to be minimized.

replace p with the reference distribution, which takes the value 1 for the correct target label y'_x and 0 for all the other target labels. Hence, we finally obtain the cross entropy loss over the X training set:

$$L(X) = -\sum_{x \in X} log(q(y'_x|x)) \tag{2.3}$$

which represents the negative log-likelihood of the training set. This means that minimizing the cross entropy loss corresponds to maximizing the likelihood of the observation in the training set.

Minimizing (or optimizing) a loss in deep learning means finding the optimal parameters $\boldsymbol{\theta}'$ that minimize it:

$$\boldsymbol{\theta}' = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L(X; \boldsymbol{\theta}) \tag{2.4}$$

As in most of the cases the problem is intractable, in practice the optimization consists in repeatedly computing the derivative of the loss function with respect to the model parameters $\frac{\partial L(X;\boldsymbol{\theta})}{\partial \theta}$, and updating the parameters by moving in the opposite sign of the derivative by a small step ϵ, known as *learning rate*:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \frac{\partial L(X; \boldsymbol{\theta})}{\partial \theta} \tag{2.5}$$

This procedure is operated for a fixed number of times N to obtain the final estimate parameters (where N and ϵ are hyperparameters), and takes the name of *gradient descent* (Cauchy, 1847).

In reality, as the amount of training data is huge, a plain application of the gradient descent is not effective because every small step requires huge computational costs (and time). For this reason, the gradient is estimated only on small subsets of the training set, known as *batches.* We refer to this practice as *stochastic gradient descent* (SGD, Bottou 1999), which is also motivated by its regularization effect (Wilson and Martinez, 2003).

The main issue of SGD is its lack of stability, as different batches may lead to contrasting movements and slow-down the overall descent. To face this problem, the concept of momentum (Polyak, 1964) has been introduced, averaging the contribution of the gradient on a batch with gradients previously computed on other batches, with different proposal on how to do so that lead to a plethora of similar optimizers (Duchi et al., 2011; Kingma and Ba, 2015; Ruder, 2016).

2.2 Deep Learning and Speech Processing

While the previous section described general concepts on neural networks, in this section we turn to its application to speech processing tasks, such as ASR and ST. First, we describe how audio and text are modeled (or represented), i.e. in which form they are described in deep learning applications (§2.2.1). Then, we introduce on sequence-to-sequence architectures (§2.2.2), with a particular focus on the widespread Transformer (§2.2.3), its adaptation for speech tasks (§2.2.4), and on the recent state-of-the-art Conformer architecture (§2.2.5).

2.2.1 Speech and Text Representation

In signal processing, audio is commonly represented as a sequence of overlapping frames with a 25ms window size and 10ms shift (Oppenheim et al., 1999). For each of the frames, a set (or vector) of features, named mel-scaled cepstral coefficients (Davis and Mermelstein, 1980), are extracted through a multistep process from the speech signal. As a result, the input to speech processing systems is a sequence of features of considerable and variable length (e.g. 10s of audio lead to a sequence of ∼1,000 vectors, one every 10ms, while for 11s the length is ∼1,100). The features are usually 40- or 80- dimensional vectors that represent the spectral envelope, which is associated with the vocal tract characteristics (Furui, 2010).

The text, instead, has been historically represented with a fixed vocabulary of known words (Sutskever et al., 2014; Bahdanau et al., 2015), typically the most frequent 30k-80k words (Collobert et al., 2011; Jean et al., 2015). This was motivated both by the intuition that words constitute a basic semantic unit (Jackendoff, 1990), and by the difficulty in effectively handling long-range dependencies with the longer sequences produced by character-level segmentation of the text, which required architectural adaptation to achieve competitive results (Costa-jussà and Fonollosa, 2016; Lee et al., 2017). Nowadays, Byte-Pair Encoding (BPE, Sennrich et al. 2016) has become the *de-facto* standard in MT (Koehn, 2017). By representing the input text as a sequence of subword units, it enables open-vocabulary translation while keeping reasonable the sequence length, achieving state-of-the-art results. The size of the vocabulary obtained with BPE is controlled by a parameter that represents the number of *merge rules* to define. The algorithm starts from the characters in the training set, then it iteratively adds a rule merging the two tokens that occur more frequently together. At the end of this process, the vocabulary is obtained as the set of possible subword units obtained by applying the learned merge rules.

2.2.2 Sequence-to-sequence Models

As we have just seen, both audio and text are represented as sequences (respectively, of vectors and subword units) with variable length. The first type of neural network capable of handling sequences of variable length has been introduced by Rumelhart et al. (1986) and takes the name of recurrent neural network (RNN).[2] An RNN can convert *i)* a sequence of vectors into a

[2] In this thesis, we do not cover the internal functioning of RNNs, as nowadays they have been superseded by the Transformer architecture (Vaswani et al., 2017) for most natural language processing (NLP) and speech processing tasks (Dong et al., 2018; Di Gangi et al., 2019c) and we never employ them. Please refer to (Goodfellow et al., 2016) for a thorough explanation of RNNs.

single vector, *ii)* a sequence of vectors into another sequence of the same size, or *iii)* a single vector into a sequence of vectors (Goodfellow et al., 2016). However, in text-to-text translation and speech-to-text applications the input sequence (of variable length) has to be converted into a sequence of a different and variable length. For this reason, encoder-decoder architectures (also known as sequence-to-sequence) were introduced (Sutskever et al., 2014; Cho et al., 2014).

In their simplest form, both the encoder and the decoder are composed by a single RNN and the information flows from the encoder to the decoder by means of a single vector (usually called *context* vector). Specifically, the encoder RNN maps the input sequence into the context vector, which represents the content of the input. The context vector is then passed to the decoder RNN that converts it into the output sequence. Formally, the decoder (D) predicts the probability distribution over all the tokens of the target vocabulary (V) given the context vector (C) and the previously generated tokens:

$$p_V(y_t) = \text{softmax}(D(C; h_{t-1})) \tag{2.6}$$

where h_{t-1} is a hidden representation that summarizes the previously generated tokens $y_0, ..., y_{t-1}$. Abstracting from the RNN case, the definition can be generalized as:

$$p_V(y_t) = \text{softmax}(D(E(X); y_0, ..., y_{t-1})) \tag{2.7}$$

where E is the encoder, and X is the input sequence. This property of relying on the previously generated tokens is what characterizes *autoregressive* models.[3] At training time, the previous tokens are drawn from the

[3] Although non-autoregressive (NAR) models have been recently proposed (Gu et al., 2018), they are not yet competitive with the autoregressive ones and, consequently, not widespread. As such, they are not covered in the thesis. For an overview of NAR solutions, the reader can refer to (Gu and Tan, 2022).

reference to parallelize the computation over the sequence (*teacher forcing* – Kremer and Kolen 2001), while at inference time they are taken from the most likely predictions according to the network. The generation process continues until it reaches a special symbol added to the target vocabulary (Schmidhuber, 2012), known as *end of sentence* (*eos*).

The main well-known limitation of RNNs is their difficulty in modeling long-term dependencies, due to vanishing and exploding gradient issues (Bengio et al., 1993, 1994; Kolen and Kremer, 2001). Indeed, RNNs process the input sequence step-by-step and each time step has access to the previous ones only through a single accumulated vector (*hidden state*). As the hidden state is updated at each time step, the gradient diminishes (up to vanishing) as the distance between two vectors increases, making it impossible for distant elements to have a significant contribution.

2.2.3 Transformer

To overcome the above-mentioned limitation of RNNs, the Transformer (Vaswani et al., 2017) architecture has been introduced. The Transformer relies on the *attention* mechanism (Bahdanau et al., 2015) that allows any vector of a sequence to "look at" (or "pay attention to") any vector of another sequence, regardless of their position. The specific attention mechanism adopted is a variant of the dot-product attention (Luong et al., 2015), that is formulated as follows:

$$\text{Attn}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.8}$$

where Q is the query, K is the key, V is the value, and d_k is the size of the dimension of the key (and query) vector. In particular, in Transformer models there are two types of attention: the *self-attention*, and the *cross-attention* (or *encoder-decoder attention*). In the first case, Q, K, and V

are all derived from the same input sequence X – which, in our case, can be either the sequence representing the speech signal or the sequence of previously generated tokens fed into the decoder – transformed by three different linear projections with learned weights W_Q, W_K, and W_V:

$$\text{SelfAttn}(X) = \text{softmax}\Big(\frac{W_Q X (W_K X)^T}{\sqrt{d_k}}\Big) W_V X \tag{2.9}$$

In the second case, instead, the attention is computed by looking at the encoder output $E(X)$, to find the information relevant for the decoding embeddings H_D:

$$\text{CrossAttn}(X, H_D) = \text{softmax}\Big(\frac{W_Q E(X) (W_K H_D)^T}{\sqrt{d_k}}\Big) W_V E(X) \tag{2.10}$$

In practice, Vaswani et al. (2017) found that it is more effective to divide the input Q, K, and V into h chunks – called *heads* – of equal size, transform each head with a dedicated weight matrix, compute the attention separately on these heads, and then concatenate the results. This operation takes the name of *multi-head attention*, and can be formulated as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Attn}(q_0, k_0, v_0), ..., \text{Attn}(q_h, k_h, v_h)) W_O \tag{2.11}$$

where $W_O \in \mathbb{R}^{(d_k, d_k)}$ is a learned matrix. Since the *multi-head attention* is always used, hereinafter the term attention indicates the multi-head variant.

Figure 2.1 depicts the whole architecture. The encoder is made of a stack of Transformer encoder layers, each of them composed of two sublayers: a self-attention, and a feed-forward network (FFN).[4] Both the sublayers are followed by a layer normalization (Ba et al., 2016), and are wrapped

[4] Also known as multi-layer perceptron (MLP), it is a stack of linear layers (2 in this case).

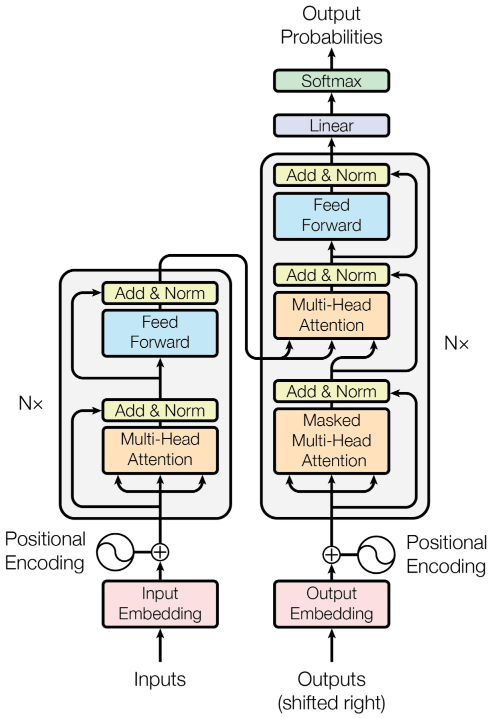


Figure 2.1: Transformer architecture. Image taken from (Vaswani et al., 2017).

by residual connections (He et al., 2016), which means that the input of the sublayer is added to its output. Similarly, the decoder is a stack of Transformer decoder layers, which are analogous to the encoder ones, but have an additional sublayer placed between the self-attention and the FFN: a cross attention that allows the decoder embeddings to look at the encoder outputs.

2.2.4 Transformer for Speech

The success of the Transformer architecture in MT, LM, and NLP (Radford et al., 2018; Devlin et al., 2019) has led to its adoption also for ASR (Dong et al., 2018). However, applying the Transformer architecture directly to speech input is not feasible, as it would require a prohibitive amount of memory. Indeed, self-attention layers have a quadratic memory complexity

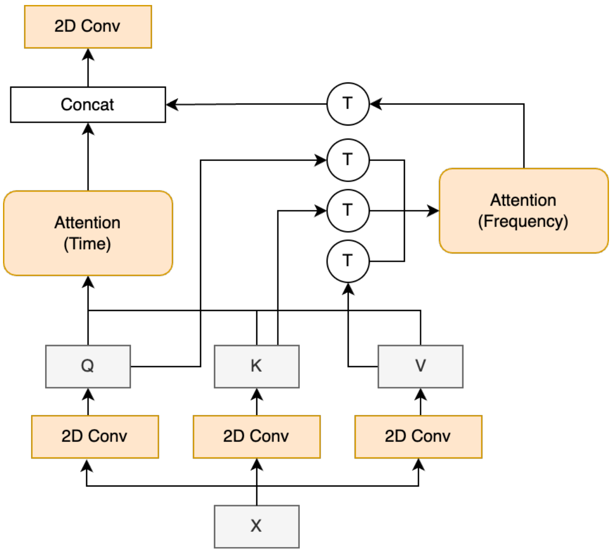


Figure 2.2: 2D self-attention. X is the input tensor representing an audio sequence (see §2.2.1 for more details), and T is the transpose operation.

in the length of the input sequence,[5] and the speech input of the network is a sequence of samples collected from a high frequency (typically one sample every 10ms, as seen in §2.2.1), usually one order of magnitude longer than the corresponding MT input sequence derived from the text. For this reason, Dong et al. (2018) reduce the length of the input sequence by a factor of 4 by means of two 2D convolutional neural networks (CNNs, LeCun 1989).[6]

In addition, Dong et al. (2018) also introduce the concept of *2D self-attention*. Their hypothesis is that the network architecture should model the varying correlations between different frequencies and time, as these are useful for humans. Figure 2.2 depicts the 2D self-attention mechanism.

[5] Q and K are tensors of size $B \times T \times C$, where B is the batch size, T is the length of the input sequence, and C is the number of features. So the product QK^T is a $B \times T \times T$ attention matrix.

[6] Convolutions apply the same operation (the multiplication with a learned tensor, named *kernel*) to a moving window of the input. The parameters that control its functions are: the size of the kernel, the stride (i.e. how much the window over the input shifts left/down each time), and the padding size. Setting the stride equal to 2 halves the shape of the input tensor over that dimension: in this way, two consecutive CNN layers with stride 2 lead to a 4 $(2 \cdot 2)$ times reduction of the input size over that dimension.

Differently from the attention introduced in §2.2.3, the input ($X \in \mathbb{R}^{T \times C}$) is transformed with 2D convolutions instead of linear projections to obtain Q, K, and V. Then, Q, K, and V are passed to two different attentions: one on the time axis, and the other on the frequency axis (in this case, Q, K, and V are transposed before being fed to the attention mechanism and then the result is transposed back).

The tensors resulting from the two attention operations are concatenated on the frequency axis. The concatenated tensor has shape $T \times 2C$ before being processed by the last 2D convolution, which has stride 2 on the frequency dimension, hence producing an output with the same shape of X.

To sum up, their encoder architecture is made of 2 CNNs that downsample the input by a factor of 4 over the time dimension, and 2 2D self-attentions followed by a Transformer encoder. The decoder is instead a plain Transformer decoder.

2.2.5 Conformer

The success of the Transformer architecture for speech processing tasks, in particular for ASR, has motivated research to encode further inductive biases[7] to improve the generalization – and in turn the performance – of the models. To date, the most fruitful of such efforts (contemporaneous to this thesis) is the introduction of the Conformer architecture (Gulati et al., 2020), which modifies the structure of the encoder layers with respect to the Transformer.

The changes introduced in the Conformer encoder layer can be summarized as follows: *i)* relative sinusoidal positional encoding (Dai et al., 2019) are introduced in the self-attention for improved generalization with respect to varying input lengths; *ii)* the FFN sublayer is replaced by two FFNs that

[7] Priors or assumptions related to how the information should be processed (Mitchell, 1980).

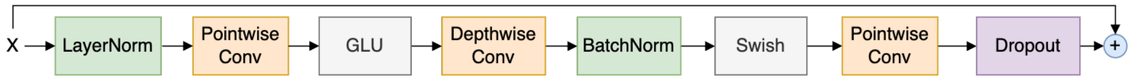


Figure 2.3: Convolutional module in the Conformer encoder layer. All convolutional blocks are 1D convolutions.

wrap the self-attention, inspired by the Macaron-Net (Lu et al., 2019a); *iii)* a convolutional module (depicted in Figure 2.3) is added immediately after the self-attention, before the last FFN module.

The convolutional module is wrapped in a residual connection. After a layer normalization, a pointwise convolution transforms each feature vector representing a time step in the same way and doubles the size of the features. The feature dimension is brought to the original size by the Gated Linear Unit (GLU) activation function (Dauphin et al., 2017). Then, a depthwise convolution with 31 kernel size is applied before a batch normalization (Ioffe and Szegedy, 2015), the Swish activation function (Ramachandran et al., 2017), and another pointwise convolution similar to the first one. At last, a dropout module (Srivastava et al., 2014) randomly masks (i.e. zeroes out) a percentage of the values to prevent the network from overfitting.

To the best of the knowledge of the author, at the time of writing this thesis the Conformer architecture represents the state of the art for speech processing, and in particular for the ASR task.

2.3 Direct Speech-to-text Translation

2.3.1 Architectures for Speech Translation

The research on direct ST systems moved its first steps with architectures based on RNNs (Bérard et al., 2016; Weiss et al., 2017). Later on, (Di Gangi et al., 2019c) demonstrated the competitiveness of the Transformer architecture for speech described in §2.2.4 also for the ST task.

Drawing on this finding, several works attempted to further improve the effectiveness of the Transformer architecture by biasing the model to attend to the local context (i.e. close vectors on the time dimension) in the self-attention of the Transformer encoder layers. (Povey et al., 2018) proposed a hard masking to restrict the span of "visible" frames only to those nearby, (Sperber et al., 2018) introduced a Gaussian distance penalty, and (Di Gangi et al., 2019b) presented a logarithmic distance penalty, which does not require any hyperparameter (the Gaussian penalty is sensitive to the initial value set for its variance) and is more effective than the Gaussian counterpart. All these works subtract the penalty to the attention weights, before the *softmax* computation, thus promoting an attention matrix with high values only around the diagonal. In mathematical terms, the self-attention mechanism in these works can be described as follows:

$$\text{SelfAttn}(X) = \text{softmax}\Big(\frac{W_Q X (W_K X)^T}{\sqrt{d_k}} - \pi(D)\Big) W_V X \tag{2.12}$$

where D is a matrix whose cells contain the distance from the diagonal cell of that row, and π is the distance penalty function.

The evolution of the architectures has moved forward during this PhD as well. To date, the initial 2D convolutions have been replaced with 1D convolutions (Wang et al., 2020b), and recent works (Inaguma et al., 2021; Vyas et al., 2021) demonstrated the superiority of the Conformer architecture in ST as well. The architectural improvements in ST contemporaneous to this PhD will be described in more details in §3, pointing out to which enhancements the work described in the thesis has contributed.

2.3.2 Overcoming Data Scarcity

Since the scarcity of available training data is one of the main issues for direct ST (as mentioned in §1.2.1), the topic has been broadly studied.

Apart from creating larger corpora (Di Gangi et al., 2019a; Wang et al., 2020a), the problem has been tackled mainly with *data augmentation* and *knowledge transfer* techniques.

Data Augmentation

The most widespread data augmentation techniques are SpecAugment, time stretch (speed perturbation), and the generation of synthetic data by augmenting the ASR corpora with translations produced by an MT system fed with the known transcripts.

SpecAugment This data augmentation technique was originally introduced for ASR (Park et al., 2019), but its effectiveness has also been demonstrated for ST (Bahar et al., 2019b). The idea is to alter the audio features that represent the speech to increase the variability of the training data and lead to more robust systems. It operates on the input features, and it consists in masking consecutive portions of the input in both the frequency and time dimensions. On every input, at each iteration, SpecAugment is applied with probability p. In case of application, it generates *frequency masking num* masks on the frequency axis and *time masking num* masks on the time axis. Each mask has a starting index, which is sampled from a uniform distribution, and a number of consecutive items to mask, which is a random number between 0 and respectively *frequency masking pars* and *time masking pars*. Masked items are set to 0.

Time stretch Nguyen et al. (2020) propose to operate directly on the features as well, with similar intents. In particular, it aims at generating the same effect of speed perturbation (Ko et al., 2015) to increase the robustness of the systems with respect to variations in speech rate. Its implementation consists in dividing the input sequence in windows of w features and re-samples each of them by a random factor s drawn by a uniform distribution (usually in the range [0.8, 1.25]). Another hyperparameter is the probability of perturbing each sample.

Synthetic translation To leverage the availability of large ASR datasets while training an ST model, parallel audio-translation pairs are created by translating the transcript of each audio sample with an MT model (Jia et al., 2019). As we discuss in §3, this method can also be considered as a knowledge transfer technique: indeed, it transfers (or distills) the knowledge of the MT model into the ST model.

Knowledge Transfer

The concept of knowledge transfer in neural networks is very similar to that of humans. Indeed, it consists in "passing" information learned by a neural network trained on a task to another neural network, addressing either the same or a different task (Gutstein et al., 2008). In direct ST, knowledge transfer from high resource tasks has been performed with model pre-training, multitask learning, and knowledge distillation (KD).

Pre-training Regarding pre-training, several studies (Bérard et al., 2018; Bansal et al., 2019) demonstrated the effectiveness of initializing the ST model encoder with that of an ASR model trained on the large ASR corpora available. Whether pre-training the decoder with that of an MT model is beneficial, instead, is controversial. In (Bahar et al., 2019a), for instance, it proved effective only with the addition of an *adapter* layer.

Multitask learning In the case of multitask learning, typically there is a single, shared encoder whose outputs are used by two separate decoders dedicated to producing respectively the transcripts (ASR) and the translations

(ST) (Weiss et al., 2017). In (Anastasopoulos and Chiang, 2018), each of these two decoders can also attend to the representations generated by the other. A slightly different approach is introduced by (Bahar et al., 2019a), which does not add an ASR decoder but relies on an auxiliary Connectionist Temporal Classification (CTC, Graves et al. 2006) loss in order to predict the transcriptions from the encoder outputs (Kim et al., 2017). The CTC algorithm enables producing an output sequence of variable length that is shorter than the input one, as in this case (the input is a long sequence of audio samples, while the output is the sequence of uttered symbols – characters, sub-words – which is significantly shorter). In particular, for each time step, the CTC produces a probability distribution over the possible target labels augmented with a dedicated `<blank>` symbol representing the absence of a target value. These distributions are then exploited to compute the probabilities of different sequences, in which consecutive equal predictions are collapsed and `<blank>` symbols are removed. Finally, the resulting sequences are compared with the target sequence.

Knowledge Distillation KD has been introduced to transfer knowledge from a big model into a small, compressed one (Hinton et al., 2015). The goal is to have a small model – named *student* in the KD learning procedure – that performs similarly to its big counterpart – named *teacher* – while being usable on low-resource devices (e.g. mobile phones). Specifically, the student is trained to learn to mimic the probability distribution of the teacher when processing the same input. This is obtained by using the probabilities generated by the teacher as reference when training the student, instead of the usual reference distribution, in which the correct label is assigned probability 1 and all the others 0. In practice, this means that the student is not trained to optimize the cross entropy loss function, but to minimize the distance between its probability distribution and the one generated by the teacher. KD has been applied to direct ST for motivations different from the original model compression. Liu et al. (2019) indeed aimed at improving the quality of an ST student model by transferring knowledge from an MT teacher, able to obtain better scores[8].

[8] Compared to MT, ST is a more complex task with lower scores, as it does not only involve translating from a source to a target language, but also recognising the speech content.

Chapter 3

Direct ST Quality and Efficiency

3.1 Introduction

When this PhD started, in 2019, the performance gap between direct and cascade systems was still large, although rapidly closing. This trend was mirrored by the findings of the International Workshop on Spoken Language Translation (IWSLT),[1] a yearly evaluation campaign where direct systems made their first appearance in 2018. On English-German, for instance, the BLEU difference between the best cascade and direct models was still substantial (1.6 points, Niehues et al. 2019), although it dropped from the 7.4 points in 2018 (Niehues et al., 2018). Such difference was even higher if we consider only the academic participants, with a 5.9 BLEU gap between the KIT cascade system and the FBK direct one. For this reason, a significant part of the PhD has been devoted to improving the overall translation quality of direct systems, so as to reach the level of performance of their cascaded counterparts.

As seen in §2.3.2, a well-known reason for the difference between the two paradigms is the limited amount of parallel corpora available for direct ST. Moreover, training a direct ST system is more difficult because the task is more complex, since it deals with understanding the content of the

[1] `http://iwslt.org`

input audio, and directly translating it into a different language without recurring to intermediate representations. This led us to focus, in the first part of the PhD, on transferring knowledge from the MT task and corpora into direct ST models (§3.3).

After this strand of activities, we focused on improving the architecture of direct ST models in terms of both efficiency and quality. Specifically, we dedicated to the design of solutions accounting for the variability over time of the amount of linguistic and phonetic information present in audio signals (e.g. due to pauses and speaking rate variations). We first introduced a dynamic content-based input compression of the audio representation based on the integration of a CTC module in the encoder (§3.4). This module aims at improving translation quality while reducing computational costs and hardware requirements both at training and inference time. Then, we worked on avoiding the initial fixed downsampling performed by state-of-the-art architectures (see §2.2 and §2.3) by proposing a new architecture (§3.5) built on an attention mechanism with reduced computational and memory requirements and complemented by the previously mentioned CTC-compression module. After the discussion of these two strands of activities, we conclude the chapter by integrating our proposals with the most recent advancements from the research community, aiming to further increase the quality and efficiency of ST models by avoiding expensive pre-trainings and filtering the training data (§3.6).

In light of the above, the contributions of this chapter include: *i)* the comparison of the sequence KD methods where an ST student learns from an MT teacher, with analysis of the problems introduced and how to solve them; *ii)* the proposal of the first dynamic sequence-length reduction for direct ST, which improves both translation quality and computational efficiency; *iii)* the first architecture for direct ST that avoids an initial fixed compression and improves audio understanding; *iv)* the demonstration that

ASR pre-training can be avoided without significant quality drops, and that a simple data filtering method based on the transcript/translation length ratio increase quality and reduces training times.

As we will see throughout this chapter, we were not alone in this chase of higher translation quality for direct – as well as for cascade – ST models. The huge amount and quality of the contributions produced by the whole research community over these three years have led to impressive results, outdating at a fast pace previous architectures, techniques, and even experimental settings, such as training parameters. For this reason, to avoid building our research on obsolete – and potentially even misleading – settings, the works we present in this chapter do not have a homogeneous setup, but each of them is represented with the best baselines and hyperparameters available at the time it was performed. This practice enforces the soundness of our overall conclusions and offers at the end of the chapter a vision of the state of the art at the time of writing this thesis. Indeed, in §3.6.4 we report the current highest result – to the best of our knowledge – obtained by a direct ST model on the popular English-German MuST-C test set without leveraging external training data (in addition to the MuST-C training set). The score (26.7) was only a pipe dream at the start of the PhD, when the best-reported result in the same condition was 17.2 (Indurthi et al., 2020).

3.2 Related Works

In this section, we provide an overview of the concepts and techniques relevant to the topics discussed in the next sections. First, we introduce the available knowledge distillation methods for sequence-to-sequence applications (§3.2.1), which we study in the context of direct ST in §3.3. Then, we discuss proposals related to the compression of the input sequences based on the amount of information brought by each item (§3.2.2), also including solutions contemporary to ours, presented in §3.4. At last, we briefly describe efficient attention mechanisms that approximate the original algorithm reducing its computational complexity (§3.2.2), as they enable processing longer input sequences, thus inspiring our architecture introduced in §3.5, the first one avoiding fixed input downsampling.

3.2.1 Sequence-to-sequence Knowledge Distillation

As seen in §2.3.2, KD has been proposed in the context of classification, where one label has to be predicted for every input. However, ST is a sequence-to-sequence task, so the output is not a single label but a sequence of variable length. Therefore, KD cannot be applied in its original form. For sequence-to-sequence tasks, Kim and Rush (2016) proposed three different techniques to distill knowledge at sequence level: *i)* word-level KD, *ii)* sequence-level KD, and *iii)* sequence interpolation.

Word-level KD (henceforth `Word-KD`) is the most similar method to the original KD definition formulated by Hinton et al. (2015) for classification. In this case, the KL divergence (see §2.1) between the teacher and student outputs is computed for every element (time-step) of the target sequence and the final distance is the sum of the divergences over all the elements (time-steps) of the sequence. Hence, the loss function is:

$$L(X) = -\sum_{x \in X} \sum_{t \in [1, len(X)]} \sum_{y \in Y} p(y_t|x, y_0, ..., y_{t-1}) * log(q(y_t|x, y_0, ..., y_{t-1})) \tag{3.1}$$

Sequence-level KD (henceforth `Seq-KD`) consists in replacing the target reference (in our case the translation provided in the training corpora) with the sequence of tokens (in our case the automatic translation) generated by the teacher model. The loss function can be either the cross entropy or one of its variants, as the label smoothed cross entropy (Szegedy et al., 2016).

Sequence interpolation (henceforth `Seq-Inter`) relies as well on the predictions of the teacher model. In this case, though, the N most likely sequences resulting from the beam search are re-scored and the one with the highest similarity with the ground truth is chosen as surrogate reference. In the case of textual outputs, such as in MT and ST, the similarity with the ground truth is computed with the BLEU score (Papineni et al., 2002).

Finally, `Word-KD` can be combined with the other two methods, resulting in two additional alternatives: `Word-KD`+`Seq-KD` and `Word-KD`+`Seq-Inter`.

In the context of direct ST, as anticipated in §2.3.2, Liu et al. (2019) train a direct ST model with `Word-KD` to transfer knowledge from the easier MT task,[2] in which models obtain better performance, and hence improve the quality of the resulting ST student model. Jia et al. (2019), instead, generate synthetic data by translating the transcripts of ASR corpora with an MT model. Although presented as a data augmentation method, this can also be interpreted as an application of the `Seq-KD` method, even though the benefits of KD cannot be isolated from those due to the additional data. However, no work investigated which is the best method to transfer knowledge from MT to ST, nor compared the above-mentioned methods, as we do in §3.3.

3.2.2 Content-based Input Compression

The information variability in speech inputs motivated the research community to find alternatives to methods that perform an initial fixed compression of the input (Sak et al., 2015; Bérard et al., 2016; Dong et al., 2018).

In the related field of ASR, Zhang et al. (2019); Na et al. (2019) propose to leverage SkipRNN (Campos et al., 2018) to dynamically decide which time steps have to be passed to the next encoder layer and which have

[2] ST does not only involve translating from a source to a target language, but also recognizing the speech content.

not. SkipRNNs determine whether a hidden state h_i, representing the time step i, should be passed to the next layer by computing an associated probability p_i of keeping it. The probability p_i is obtained by summing the cumulated probability of the previous steps c_i with an increment of probability of the current time step Δp_i, estimated from the current hidden state by means of an FFN. When p_i surpasses a threshold (0.5 in Campos et al. 2018), the hidden state h_i is passed to the next layer and c_i is reset to 0; otherwise, h_i is discarded. This method is hardly applicable, though, to Transformer architectures. Indeed, replacing the initial convolutions with a SkipRNN has two main problems: *i)* the network optimization is extremely challenging due to vanishing/exploding gradient, preventing the successful convergence of such architecture in our experiments; *ii)* as the computation in RNNs is not parallelizable, the architecture training is extremely slow.

In ST, Salesky et al. (2019) demonstrated that a phoneme-based compression of the input frames yields significant gains compared to fixed length reduction. Phone-based and linguistically-informed compression also proved useful in the context of visually grounded speech (Havard et al., 2020). Zhang et al. (2020), instead, showed that selecting a small percentage (∼16%) of the hidden states produced by a pre-trained ASR encoder according to their informativeness improves ST quality. However, these approaches respectively necessitate of a separate model to perform phoneme classification and of a pre-trained adaptive feature selection layer on top of a pre-trained ASR encoder. So, they: *i)* are affected by *error propagation* (Salesky and Black 2020 show in fact that lower quality in phone recognition significantly degrades final ST performance), *ii)* have a more complex architecture, and *iii)* at least in the case of (Salesky et al., 2019) require longer inference time, i.e. higher latency.

In contemporaneity and similarly to our work (see §3.4), Liu et al. (2020) conceptually divide the encoder of the ST model into two parts: an acoustic and a semantic encoder. On top of the first part, the acoustic encoder, a CTC loss predicts the audio transcript. This loss is not only used as an auxiliary loss to help model convergence (see §2.3.2); its predictions also determine which hidden states are passed to the semantic encoder, and which should be discarded. Namely, all hidden states corresponding to `<blank>` predictions are ignored, as well as those corresponding to a prediction equal to the previous one. The remaining hidden states are passed to the semantic encoder, which therefore receives a shorter sequence, of similar length to that of the textual representation of the utterance. The main difference between this solution and ours, as we will see in §3.4, lies in how the CTC predictions are leveraged to compress the sequence.

3.2.3 Efficient Attention

Another strand of research focused on improving the efficiency of Transformer-based architectures by reducing the computational complexity of the self-attention (Tay et al., 2020). Among others, Beltagy et al. (2020); Choromanski et al. (2021); Wang et al. (2020c); Katharopoulos et al. (2020); Zheng et al. (2022) proposed approximated attention computations with linear complexity on the length of the input sequence. Most of them, though, are complex to adopt in a scenario in which the variability of the length of the input sequence is high. For instance, low-rank approximations of the attention matrix (Wang et al., 2020c) map the K and V matrices obtained from each input sequence of any length into a fixed-length sequence by applying a linear projection. On one side, mapping those sequences to a fixed dimension can cause an excessive information loss, with a consequent performance drop. On the other, it poses technical issues: the linear projection matrix has size $n \times k$, where n is the maximum input length and k is the fixed dimension. If the input has a length n' shorter than n, which is a common case in ST and ASR due to the high variability in length of audio sequences, only the first n' weights of the matrix are updated. This results in gradients of different dimensions across GPUs, leading to training failures due to inconsistencies.

In ASR, Burchi and Vielzeuf (2021); Kim et al. (2022); Andrusenko et al. (2022) instead do not operate directly on the self-attention mechanism, but insert pooling layers to reduce the sequence length at different layers of their revisited Conformer encoders. These recent works aim at both increasing efficiency and overall quality. Their effectiveness in ST, though, has not been investigated yet.

In ST, Alastruey et al. (2021) applied the Longformer (Beltagy et al., 2020) – an architecture that features a local attention, in which each time step can attend only to those within a fixed window size – but obtained a degradation in translation quality. Alastruey et al. (2022) avoid the quality drop by computing the full self-attention in the first three layers and by applying a local attention with variable window size according to the layer and language pair. None of the above-mentioned solutions (in ASR and ST) avoid the initial sequence-length reduction performed by the convolutional layers pre-pended to Transformer/Conformer encoders for speech (see §2.2) or substitute it with a content-based compression as we do in §3.5 and §3.6. To the best of our knowledge, the only other attempt to do so is a recent work by Tsiamas et al. (2022a), in which the authors show that a Perceiver encoder (Jaegle et al., 2021) fed with the full-length input sequence scores results similar to a Transformer baseline with reduced computational cost.

3.3 Knowledge Distillation in ST

As anticipated in §3.1, the beginning of the PhD was dedicated to the study of KD, which represents one of the most promising approaches to transfer knowledge from MT to ST models. Indeed, Liu et al. (2019) showed that using an MT system as teacher brings significant improvements to direct ST models. However, as mentioned in §3.2.1, they consider only the `Word-KD` method, disregarding the other solutions to distill knowledge in a sequence-to-sequence task like ST.

In addition to filling this gap by studying which KD method is more effective in ST (§3.3.3), in this section we explore methods to improve the computation of `Word-KD` both in terms of computational efficiency and quality (§3.3.1). Lastly, we investigate the negative effects of KD and the relationship between the quality of the teacher and that of the resulting student (§3.3.4). Altogether, this investigation leads to the following overall findings: *i)* the best training recipe involves a word-level KD training followed by a fine-tuning step on the ST task, *ii)* word-level KD from MT can lead to the omission of all the sentences after the first one in multi-sentential utterances (though these problems are alleviated by the fine-tuning on the ST task), and *iii)* the quality of the ST student model strongly depends on the quality of the MT teacher model, although the correlation is not linear.

3.3.1 Efficient Word-level KD

The definition of the `Word-KD` method exposed in §3.2.1 implies that the whole output distribution of the teacher model is compared with the whole output distribution of the student at each decoding step. In practice, this is highly inefficient since pre-computing and storing the output probabilities for each token of each sequence requires huge storage capacity (e.g. with ∼100,000 samples of average length 100 and 8,000 labels in the output distribution, we would need to store 80,000,000,000 floats, corresponding to more than ∼320 GB of storage). On the other hand, re-computing the teacher target label at every iteration entails a forward pass on the teacher network for every input batch, leading to a significant increase in the training time.

Considering that the $softmax$ operation produces peaky outputs that tend to concentrate most of the probability distribution across up to 3-4 tokens, we hypothesize that truncating the probability distribution and reducing the loss computation to only the K most likely labels can speed up the training without compromising the quality of the resulting model.

Moreover, as mentioned in §3.2.1, KD has been proposed with a hyper-parameter, the *temperature*, that controls the smoothness of the output distribution and increases/decreases the importance of the so-called *dark knowledge*. As previous work on the topic (Liu et al., 2019) disregarded this aspect, we fill the gap by exploring whether favoring the learning of such *dark knowledge* leads to better results.

3.3.2 Experimental Settings

In this section, we provide a comprehensive description of the data, architectures, and parameters utilized in our experiments for the MT teachers, the ST models, and the ASR models employed for pre-training the encoder of ST systems.

Data and Evaluation

For the initial comparisons, we train and evaluate systems in a controlled setting, using only the data from Librispeech (Panayotov et al., 2015), which contains $132,553$ *(audio, transcript, translation)* triplets for the English→French language direction.

When validating our findings in high-resource conditions, instead, we use the following MT, ASR, and ST corpora for three language pairs: English→{French, German, Italian}. The MT data is a selection of the OPUS corpora (Tiedemann, 2016), filtered using the cleaning utilities of ModernMT (Bertoldi et al., 2017). OPUS contains parallel sentences automatically extracted from the web. As such, their nature is very different from the ASR and ST data, which is based on recorded sessions (TED or European Parliament talks) or book/manual readings and whose utterances can contain more than one sentence. The ASR data include How2 (Sanabria et al., 2018), Librispeech (Panayotov et al., 2015), Mozilla Common Voice,[3] TED-LIUM 3 (Hernandez et al., 2018), and MuST-C (Di Gangi et al., 2019a), which also constitutes our ST corpus with Europarl-ST (Iranzo-Sánchez et al., 2020).

The input audio is pre-processed by extracting 40 features using Mel filter bank with overlapping windows of 25 ms and 10 ms step size. The extracted features are then normalized per speaker. This pre-processing is performed with XNMT (Neubig et al., 2018). Samples resulting in more than 2,000 vectors (i.e. longer than 20s) are discarded to avoid excessive memory requirements at training time. Both ASR and ST trainings augment source audio with SpecAugment, using 0.5 as probability, 13 as *frequency masking pars*, 20 as *time masking pars*, 2 as *frequency masking num*, and 2 as *time masking num*. Text, instead, is tokenized after punctuation normalization with Moses (Koehn et al., 2007) and segmented into sub-word units using 8,000 BPE merge rules (Di Gangi et al., 2020a) jointly learned on the two languages of the MT dataset.

We evaluate the systems on the MuST-C test sets. The translation quality of the systems is assessed with BLEU, using the `multi-bleu.pl` script.

Architectures

Our MT models are Transformer models with 6 encoder layers and 6 decoder layers. We use a small model with 512 hidden features and 8 attention

[3] `https://voice.mozilla.org/`

2D Self-Attention	Encoder	Decoder	BLEU
2	6	6	16.50
0	8	6	**16.90**
2	9	6	17.08
2	9	4	17.06
2	12	4	**17.31**

Table 3.1: Results on Librispeech with Word KD varying the number of layers.

heads in all attention layers and 1,024 hidden features in the FFNs of the Transformer layers. In the experiments involving a larger amount of data, all these hyperparameters are doubled.

In ASR and ST, the input features are processed with two 2D convolutions, each having stride 2, that reduce the sequence length by a factor of four. This sequence is then fed to the Transformer encoder, whose self-attention layers are modified by biasing the attention matrix toward close elements with a logarithmic distance penalty. We use a small model, with 256 hidden features and 4 attention heads in all attention layers and 1,024 hidden features in the FFNs of Transformer layers. The number of Transformer encoder layers is 8 and the number of Transformer decoder layers is 6. In the high-resource experiments, we use 11 Transformer encoder layers and 4 Transformer decoder layers for our ST models, while the ASR models used for the pre-training have 8 Transformer encoder layers and 6 Transformer decoder layers. When loading the pre-trained encoder layers, the additional 3 layers of the ST model are randomly initialized and behave as adapter layers (Jia et al., 2019; Bahar et al., 2019a). Moreover, we increase the size of the models that have 512 hidden features and 8 attention heads in the attention layers, and 2,048 hidden features in the FFNs.

These choices are motivated by preliminary experiments on Librispeech (Panayotov et al., 2015) reported in Table 3.1, in which we observed that replacing 2D self-attention layers with additional Transformer encoder layers was beneficial to the final score. Moreover, we noticed that the addition of encoder layers improves the results, while the removal of two decoder layers does not significantly degrade the performance.

Training Details

In all our trainings we choose Adam using betas $(0.9, 0.98)$ as optimizer and, in case the loss is not the KL-divergence, we use label smoothing with smoothing factor 0.1. For ASR, the objective function also includes CTC loss, which is summed to the cross entropy. The CTC is computed on the encoder output (with the transcripts as target), and its role is only to aid model convergence and improve the final quality of the model. In all trainings, the learning rate is increased linearly for $4,000$ updates, up to the value of $5 * 10^{-3}$, and then decays with the inverse square root policy. In the fine-tunings, instead, the learning rate is kept fixed and is $1 * 10^{-4}$. The dropout is set to 0.2.

Each mini-batch contains 8 samples and we train on 8 K80 GPUs, but parameter updates are delayed after 8 mini-batches to reach an overall batch size of 512.

3.3.3 Results

First of all, we report preliminary experiments to define the best values for the K elements to keep from the teacher distribution, and the temperature T when performing `Word-KD`, as per §3.3.1. Then, we compare the three KD methods described in §3.2.1 on the Librispeech corpus. Within this controlled setting, the benefits brought by KD to the ST students are not due to the indirect exposure to additional MT data, but to the easiness to learn by extracting knowledge from the better-performing MT teacher. At

Top K	BLEU
4	16.43
8	**16.50**
64	16.37
1024	16.34

Table 3.2: Results on Librispeech with different K values, where K is the number of tokens considered for `Word-KD`.

last, we validate the effectiveness of the best method in the more realistic high-resource conditions.

Word-KD Computation

Table 3.2 reports the results for different K values. As the output is required to be a valid probability distribution, after the truncation the probabilities are re-scaled to sum up to 1. As per the formulated hypothesis based on the $softmax$ behavior, limiting the KL-divergence computation to a small number of labels does not impact performance. On the contrary, the best result is obtained with 8 labels, in line with similar findings for MT (Tan et al., 2019). Indeed, predictions with very low probabilities are likely to be uninformative and noisy and do not carry useful information about the internal knowledge of the teacher. In light of these results, hereinafter all experiments with `Word-KD` assume that the KL-divergence is only computed by setting $K = 8$, i.e. on the top 8 output labels of the teacher distribution.

Moving to the assessment of the best value to use for the temperature, instead, Table 3.3 shows that the best BLEU score is achieved by setting the temperature to 1.0, which means by training without any smoothing factor. This finding suggests that ST models – as they need to learn a more complex task – have a limited capacity with respect to MT models, and therefore focusing only on the mode of the MT model distributions is more convenient. Accordingly, in the following experiments we do not

T	BLEU
1.0	**16.50**
4.0	16.11
8.0	14.27

Table 3.3: Results on Librispeech with different temperatures (T). All differences are statistically significant with $p = 0.05$.

	BLEU
Baseline	9.4
`Word-KD`	16.5
`Seq-KD`	13.4
`Seq-Inter`	13.3
`Seq-KD` + `Word-KD`	15.7
`Word-KD` + FT `Seq-KD`	16.7†
`Seq-KD` + FT `Word-KD`	**16.8**†
`Word-KD` + FT w/o KD	**16.8**†

Table 3.4: Results of the small model on Librispeech with different KD methods and combining them in a single training or in consecutive trainings through a fine-tuning (FT). "†" indicates that improvements over `Word-KD` are statistically significant with $p = 0.05$.

apply smoothing, by setting the temperature hyperparameter to 1.0.

Word-KD, Seq-KD, Seq-Inter and their Combination

We now compare the standard cross entropy loss – which we consider our baseline – with the KD methods. The comparison is also carried out by considering different combinations of such techniques. These can be performed in two ways: by applying both techniques together in the same training, or by first training with one technique and then fine-tuning the resulting ST model with the other. We also experimented with fine-tuning (FT) without KD after the application of a KD method. The results are reported in Table 3.4.

Looking at the Baseline and the three KD techniques, we can conclude that all KD methods improve significantly over the Baseline, with gains that range from 3.9 to 7.1 BLEU points. Moreover, `Word-KD` is a clear winner among them, with a 3.1 BLEU margin over `Seq-KD`. Combining `Word-KD` and `Seq-KD` in a single training (`Seq-KD` + `Word-KD`) does not bring advantages; conversely, the result is worse (-0.8 BLEU) than the training with only `Word-KD`. The quality of the resulting model is instead improved when `Word-KD` and `Seq-KD` are applied sequentially, i.e. when a first training with either of them is followed by a fine-tuning with the other (see `Word-KD` + FT `Seq-KD` and `Seq-KD` + FT `Word-KD`). Both solutions yield small gains of 0.2-0.3 BLEU points over the `Word-KD` method alone. The same result is also obtained when training on `Word-KD` and fine-tuning on the ground truth references with label smoothed cross entropy, i.e. without KD (`Word-KD` + FT w/o KD).

Although they are in line with previous work on KD for ST from MT (Liu et al., 2019), our results do not confirm the trends shown in (Kim and Rush, 2016), where KD is used to compress MT models. Indeed, in our case `Word-KD` is a clear winner. This suggests that the effectiveness of different KD methods in a sequence-to-sequence scenario varies depending on the peculiarities of the task.

High Resource Conditions

Once defined the best KD practice in controlled settings with the above experiments, we validate its effects in the more realistic high-resource scenario, in which large parallel MT corpora are available, together with a considerable amount of speech hours with the corresponding transcripts (ASR data).[4] The ST training is carried out in three phases: *i)* a training with `Word-KD` on the ASR corpora, whose transcripts are translated into the target language with the MT model (i.e., a `Word-KD` + `Seq-KD` training

[4] Although large ST corpora are not available, plenty of ASR and MT data can be collected to build models for real use cases.

Language Pair	MT Teacher	ST after `Word-KD` (step *ii)*)	ST after fine-tuning (step *iii)*)
en-de	32.1	25.8	27.6
en-fr	46.0	36.5	40.3
en-it	32.7	22.8	27.7

Table 3.5: Scores of the MT teachers and ST students on the MuST-C tst-COMMON set for en→{fr,de,it}.

on the ASR data); *ii)* a fine-tuning with `Word-KD` on the ST corpora; *iii)* a fine-tuning without KD, as per the best training method in our previous experiments. The ST encoder is initialized with that of an ASR model trained on the above-listed corpora and scoring 10.2 WER on the MuST-C test set.

Table 3.5 reports the scores of the MT teachers, the ST students after the first two training steps (those including `Word-KD`), and the final ST score after the last fine-tuning without KD. These results emphasize the importance of the last fine-tuning without KD to obtain state-of-the-art results. Indeed, we can see that in the real scenario, where there is a significant domain mismatch between the MT and the ST training data (web-crawled pairs vs TED talks, see §3.3.2), distilling the MT knowledge brings information and benefits that mostly emerge in the overall scores after the final fine-tuning. Our hypothesis is that the additional useful knowledge is counterbalanced by the negative effect of learning patterns that are valid only for the MT training data. In the following, we study what these spurious patterns and negative effects are.

3.3.4 Analysis

In this section, we first investigate which are the possible negative effects introduced by learning from an MT teacher. Then, we explore how important is the quality of the MT teacher for the ST student.

KD Negative Effects

We conducted a manual analysis on the en-it outputs, as en-it shows the highest gain (+4.9 BLEU, while en-fr has a +3.8 BLEU and en-de a +1.8 BLEU improvement – see Table 3.5). In particular, we selected and inspected the samples with the highest TER (Snover et al., 2006) gains after fine-tuning. This analysis revealed two main types of output improvements.

Avoid Sentence Omissions. The ST student often generates only the first sentence of an utterance and terminates the generation after it, regardless of whether the utterance really contains a single sentence or more than one. In this second case, hence, the output turns out to be truncated. Most likely, the root cause can be attributed to the nature of the data the MT teacher is trained on: indeed, MT corpora contain mostly parallel sentences and rarely a sample contains more than one sentence. As such, the MT teacher (and, in turn, its ST student) learn to terminate the sentence after the full stop. Fine-tuning on the ST task, however, solves the issue: upon manual inspection, none of the outputs of the fine-tuned model is affected by omissions.

Verbal Tense and Lexical Choices. The ST student often chooses verbal tenses that are more common and less accurate. For instance, *"That meant I was going to be on television"* has been translated by the ST student as *"Questo significava che* ***stavo andando*** *in tv"*. Although it might be considered acceptable in a colloquial scenario, this translation is grammatically wrong as the imperfect indicative verbal tense (*stavo andando*) should not be used in objective prepositions referring to past events. The fine-tuned model, instead, produces the correct translation with the grammatically-correct verbal tense *"Questo significava che* ***sarei andata*** *in televisione"*. Similarly, in some cases the ST student prefers common, generic words. For instance, *"She has taken a **course** in a **business school**, and she has become a veterinary doctor"* should be translated as *"Ha seguito un **corso** in una **scuola di business**, ed è diventata una veterinaria"*. However, the ST student produces *lezione* (lesson) instead of *corso* and *economia* (economics) instead of *scuola di business*. After fine-tuning, the model uses the correct terms *corso* and *business school*. Though important in terms of final score, these improvements may be also considered as an adaptation to a different domain and linguistic style (less colloquial), mostly due to the domain mismatch between the MT training data (web-crawled sentence pairs) and the ST data (TED talks).

As mentioned, the fine-tuning enhancements are mostly adaptations to the ST data and domain, which have peculiarities that differentiate them for the MT corpus used to train the MT teacher. This explains also the reason why the gains obtained with the fine-tuning are smaller in §3.3.3, where the MT and ST data coincide.

The Importance of Teacher Quality

So far, we analyzed *what* the ST student learns from the MT teacher. However, we have not yet addressed the question: *how much* does the ST student learn from the MT teacher? How important is the quality of the MT teacher for the ST student quality? To answer these questions, we experimented using MT teachers of different quality (controlled by adding/removing data) to train ST students on the MuST-C en-it section, the same used in our previous analysis. We tested both `Word-KD` and `Seq-KD` to understand whether the quality of the teacher is a factor to be considered when choosing the KD method, e.g. whether with low-performing teachers one method is preferable, while with strong teachers the other one is superior.

We consider four teachers with different quality levels and the resulting

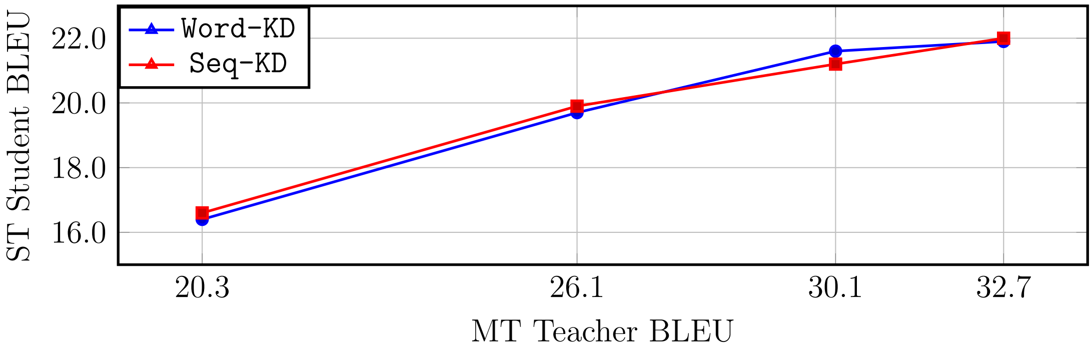


Figure 3.1: ST student performance (y axis – BLEU score) according to the MT teacher quality (x axis – BLEU score), when using `Word-KD` and `Seq-KD`, on MuST-C en-it.

teacher quality is controlled by sampling the training data. In particular, the best teacher (scoring 32.7 BLEU) is trained on the whole OPUS corpus (60M sentence pairs). Then 10M, 1M and 250K (the size of the MuST-C dataset) sentences are sampled to define the training sets for the other three teachers, ensuring that all the sentences included in one training set are also present in the bigger datasets. The teachers trained on these smaller datasets score respectively 30.1, 26.1, and 20.3 BLEU. Unsurprisingly, the score of the MT system trained on the MuST-C dataset (28 BLEU) is significantly higher than the results of the MT models trained on a similar amount of out-of-domain data. Indeed, we need to increase by 40 times the size of the training data to obtain better scores. Although the scores are relatively low, this represents a normal working condition when using KD as a source of potentially useful external knowledge, as MT models are usually trained on large generic training corpora.

Looking at Figure 3.1, we can confirm the intuition that a better teacher leads to a better student, although the students' training set is the same and the margin with the teacher is huge even with the worst teacher (+3.7 BLEU). In addition, we can notice that the student is able to only partially learn the additional knowledge of the teacher: the gap between the MT teacher and the ST student's quality increases with the teacher quality and the ST student BLEU score has not a linear dependency with the teacher BLEU, as the benefits become smaller at higher BLEU scores (the `Word-KD` student gains only 0.3 BLEU when the teacher improves from 30.1 to 32.7). We can conclude that the student is able to learn only part of the teacher knowledge and the lower scores are not only due to a lower capacity of the student model, since the student has a large margin of improvement even with bad teachers, but improves significantly with a better teacher.

Finally, the comparison of `Word-KD` and `Seq-KD` results in similar trends and scores. The two methods behave similarly both with low and high-quality teachers and they show the same performance. Indeed, the very small BLEU differences can be ascribed to statistical fluctuations and one method is not always better than the other. These results do not confirm the superiority of `Word-KD` shown in §3.3.3, but the difference can be explained with the different setting and scenario: in §3.3.3 the training set of the MT teacher is the same set on which the ST student is trained, while here the MT teacher is trained on different, out-of-domain corpora.

All in all, this analysis indicates that only part of the knowledge of the teacher can be learned by the student. Future research might try to explain which information can be learned by the student to provide insights on methods to create models that are better teachers as they focus on what can be learned by student models or to understand how to inject into the student the knowledge of the teacher that current KD methods do not allow learning.

3.3.5 Summary

Our quest for high-quality direct ST systems started from a systematic analysis of the application of KD techniques to transfer the knowledge of an MT model into an ST system. First, we compared the methods proposed in literature to distill knowledge in sequence-to-sequence models, as MT and ST systems are. Our experiments, besides confirming the benefits brought by KD, have shown the superiority of the `Word-KD` technique and the importance of fine-tuning the resulting ST student on the ST data without KD. Second, we individuated the main limitations introduced by distilling knowledge from an MT teacher: sentence truncation and omission in multi-sentential utterances. We also showed that these issues can be overcome with a simple fine-tuning without KD. Third, we demonstrated that the quality of the MT teacher is essential and that a better MT teacher leads to a better ST student, although the student gains tend to saturate when the teacher scores are high. Overall, our results show that distilling knowledge from MT is a good knowledge transfer technique, which enables benefiting from the abundance of parallel textual data in the ST task. However, it requires some adroitness, as shown by the importance of a KD-independent fine-tuning to solve the undesirable side effect of learning behaviors of the MT teacher that can be harmful to the task at hand. The rest of the chapter is dedicated to improving the quality of ST systems focusing on a different aspect. Specifically, it describes our proposal of encoder architectures that account for the variability of the amount of information in the audio signal by compressing the input sequence with a dynamic, content-based method.

3.4 CTC Compression

As seen in §3.2.2, previous studies have demonstrated that a dynamic phone-informed compression of the input audio is beneficial for ST. However, none of them tested this solution in a direct ST system, in which a single model translates the input audio into the target language without intermediate representations. Here, we propose the first method able to perform a dynamic compression of the input in direct ST models (§3.4.1). In particular, we exploit the CTC to compress the input sequence according to its phonetic characteristics, i.e. the corresponding phones obtained from a phonetic conversion of the transcripts (§3.4.2). As we will see, our experiments (§3.4.4) demonstrate that our solution yields an improvement of up to 1.5 BLEU over a strong baseline on two language pairs (English-Italian and English-German), contextually reducing the memory footprint by more than 10%.

3.4.1 Architecture

The CTC algorithm (see §2.3.2) is usually employed for training a model to predict an output sequence of variable length that is shorter than the input one. This is the case of speech/phone recognition, as the input is a long sequence of audio samples, while the output is the sequence of uttered symbols (e.g. phones, sub-words), which is significantly shorter. In particular, for each time step, the CTC produces a probability distribution over the possible target labels augmented with a dedicated `<blank>` symbol representing the absence of a target value. These distributions are then exploited to compute the probabilities of different sequences, in which consecutive equal predictions are collapsed and `<blank>` symbols are removed. Finally, the resulting sequences are compared with the target sequence.

As seen in §2.3.2, adding an auxiliary CTC loss to the training of direct ST and acoustic ASR models has been shown to improve performance. In (Kim et al., 2017; Bahar et al., 2019a), the CTC loss is computed against the transcripts on the encoder output to favor model convergence. Generally, the CTC loss can be added to the output of any encoder layer, as shown in Figure 3.2, where the hyperparameter N_{CTC} indicates the number of the layer at which the CTC is computed. Formally, the final loss function is:

$$\lambda = CTC(E_{\text{N}_{\text{CTC}}}) + CE(D_{\text{N}_{\text{D}}}) \tag{3.2}$$

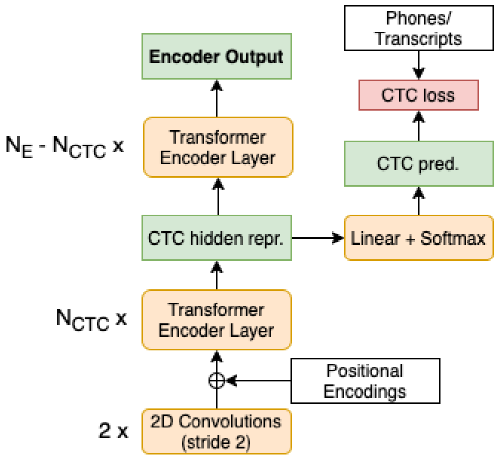


Figure 3.2: Encoder architecture with CTC loss.

where E_{x} is the output of the x-th encoder layer, $D_{\mathrm{N_D}}$ is the decoder output, CTC is the CTC function, and CE is the label smoothed cross entropy. If N_{CTC} is equal to the number of encoder layers (N_{E}), the CTC input is the encoder output. In our experiments (§3.4.4), we consider this solution as our baseline and we also test it with phones as target.

As shown in Figure 3.2, we use as model a Transformer, whose encoder layers are preceded by two 2D convolutional layers that reduce the input size by a factor of 4. Therefore, the CTC produces a prediction every 4 input time frames. The sequence length reduction is necessary both because it makes possible the training (otherwise out-of-memory errors would occur) and to have a fair comparison with modern state-of-the-art models. A logarithmic distance penalty is added to all the Transformer encoder layers.

Our proposed architecture is represented in Figure 3.3. The difference with the baseline is the addition of a block (*Collapse same predictions*) that exploits the CTC predictions to compress the input elements (vectors). Therefore, in this case the CTC does not only help model convergence, but it also identifies variable-length segments representing the same content. In this way, dense audio portions can be given more importance, while redundant/uninformative vectors can be compressed. This allows the

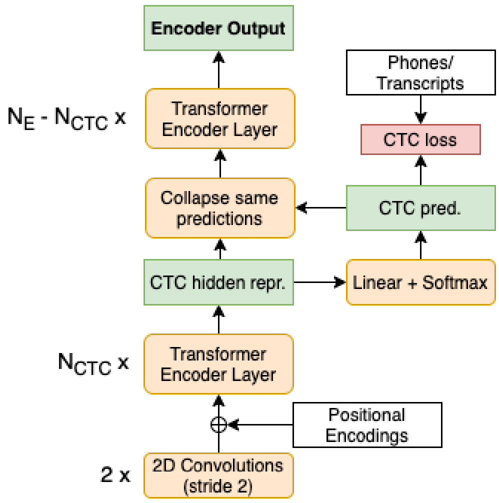


Figure 3.3: Encoder architecture with CTC compression.

following encoder layers and the decoder to attend to useful information without being “distracted” by noisy elements. The architecture is a direct ST solution as there is a single model whose parameters are optimized together without intermediate representations. At inference time, the only input is the audio and the model produces the translation into the target language (contextually generating the transcripts/phones with the CTC).

We compare three techniques to compress the consecutive vectors with the same CTC prediction:

- **Average.** The vectors to be collapsed together are averaged. As there is only a linear layer between the CTC inputs and its predictions, the vectors in each group are likely to be similar, so the compression should not remove much information.

- **Weighted.** The vectors are averaged, but the weight of each vector depends on the confidence (i.e. the predicted probability) of the CTC prediction. This solution is meant to give less importance to vectors whose phone/transcript is not certain.

- **Softmax.** In this case, the weight of each vector is obtained by computing the `softmax` of the CTC predicted probabilities. The idea is to propagate information (nearly) only through a single input vector (the more confident one) for each group.

3.4.2 Data

We experiment on the English-Italian (465 hours) and English-German (408 hours) sections of MuST-C. For each set (train, validation, test), it contains the audio files, the transcripts, the translations and a YAML file with the start time and duration of the segments.

In addition, we extract the phones using Gentle.[5] Besides aligning the transcripts with the audio, Gentle returns the start and end time for each recognized word, together with the corresponding phones. For the words not recognized in the audio, Gentle does not provide the phones, so we lookup their phonetic transcription on the VoxForge[6] dictionary. For each sample in the corpus, we rely on the YAML file and the alignments generated by Gentle to get all the words (and phones) belonging to it. The phones have a suffix indicating the position in a word (at the end, at the beginning, in the middle or standalone). We also generated a version without the suffix (we refer to it as `PH W/O POS`). The resulting dictionaries contain respectively 144 and 48 symbols.

3.4.3 Experimental Settings

We evaluate performance on MuST-C with WER for ASR and with BLEU (Papineni et al., 2002) computed with `multi-bleu.pl`[7] and SacreBLEU (Post, 2018)[8] for ST.

[5] `https://lowerquality.com/gentle/`

[6] `http://www.voxforge.org/home`

[7] To be comparable with previous works.

[8] The version signature is: `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.3`.

	WER (↓)	**RAM (MB)**
Baseline - 8L EN	16.0	6929 (1.00)
8L PH	**15.6**	6661 (0.96)
2L PH AVG	21.2	3375 (0.49)
4L PH AVG	17.5	4542 (0.66)
8L PH AVG	16.3	6286 (0.91)
8L PH W/O POS. AVG	16.4	6565 (0.95)
8L EN AVG	16.3	6068 (0.88)

Table 3.6: Results on ASR using the CTC loss with transcripts and phones as target. `AVG` indicates that sequence is compressed by averaging the vectors.

The ST and ASR architectures have the same hyper-parameters of the big models described in §3.3.2. We use 8 Transformer encoder layers and 6 decoder layers for ASR and 11 encoder and 4 decoder layers for ST unless stated otherwise. Training details are also similar, including the batch size, the GPUs, and the preprocessing steps used to extract the features from the input audio. We change only the policy of the learning rate, which linearly increases from 3e-4 to 5e-3 for 4,000 updates, then decays with the inverse square root. The text is tokenized into subwords with 1,000 BPE merge rules. We train until the model does not improve on the validation set for 5 epochs and we average the last 5 checkpoints.

3.4.4 Results

As our CTC-compression mechanism is applicable in all tasks where the source is speech, we evaluate its effectiveness both in the ASR and direct ST tasks.

ASR

We first tested whether ASR benefits from the usage of phones and sequence compression. Table 3.6 shows that having phones instead of English transcripts (Baseline - 8L EN) as target of the CTC loss (`8L PH`) without

	en-it			**en-de**		
	BLEU	**SacreBLEU**	**RAM (MB)**	**BLEU**	**SacreBLEU**	**RAM (MB)**
1. Di Gangi et al. (2020a)	20.1	-	-	19.1	-	-
2. Baseline - 8L EN	22.1	21.8	9624 (1.00)	20.4	20.5	9166 (1.00)
3. 8L PH	22.6*	22.3*	9567 (0.99)	21.6*	21.6*	9190 (1.00)
4. 2L PH AVG	20.2	20.0	5804 (0.60)	17.8	17.8	4484 (0.49)
5. 4L PH AVG	21.6	21.3	6193 (0.64)	20.1	20.2	5186 (0.57)
6. 8L PH AVG	*23.2*†	*22.8*†	8554 (0.89)	*21.8**	*21.9**	7348 (0.80)
7. 8L PH WEIGHTED	22.7*	22.5*	7636 (0.79)	21.7*	21.8*	7380 (0.81)
8. 8L PH SOFTMAX	22.6*	22.3*	7892 (0.82)	*21.8**	*21.9**	7436 (0.81)
9. 8L PH W/O POS. AVG	22.2	22.0	7451 (0.77)	21.5*	21.6*	7274 (0.79)
10. 8L EN AVG	22.2	21.9	8287 (0.86)	20.6	20.7	7143 (0.78)
11. 8L PH AVG (14+6L)	**23.4**†	**23.2**†	8658 (0.90)	**21.9**†	**22.0**†	7719 (0.84)

Table 3.7: Results using the CTC loss with transcripts and phones as target. AVG, WEIGHTED and SOFTMAX indicate the compression method. BLEU refers to scores computes with *multi-bleu.pl* for the sake of comparison with previous work. If none is specified, no compression is performed. The symbol "*" indicates statistically significant gains with respect to the baseline. "†" indicates statistically significant gains with respect to 8L PH. Statistical significance is computed according to (Koehn, 2004) with $\alpha = 0.05$. Scores in *italic* indicate the best models among those with equal number of layers.

compression is beneficial. When compressing the sequence, there is little difference according to the target used (8L PH AVG, 8L PH W/O POS. AVG, 8L EN AVG). However, the compression causes a 0.3-0.5 WER performance degradation and a 12-5% saving of RAM. Moving the compression to previous layers (4L PH AVG, 2L PH AVG) further decreases the output quality and the RAM usage. We can conclude that compressing the input sequence harms ASR performance, but might be useful if RAM usage is critical and should be traded off with performance.

Direct ST

In early experiments, we pre-trained the first 8 layers of the ST encoder with that of the ASR model, adding three *adapter* layers (Bahar et al., 2019a). We realized that ASR pre-training was not useful (probably because ASR and ST data are the same), so we report results without pre-training.

As we want to ensure that our results are not biased by a poor baseline, we compare with (Di Gangi et al., 2020a), which uses the same framework and similar settings. As shown in Table 3.7, our baseline (`8L EN`) outperforms (Di Gangi et al., 2020a) by 2 BLEU on en-it and 1.3 BLEU on en-de.

As in ASR, replacing the transcripts with phones as the target for the CTC loss (`8L PH`) further improves respectively by 0.5 and 1.2 BLEU (see rows 2-3). We first explore the introduction of the compression at different layers (rows 4-6). Adding it to the 8th layer (`8L PH AVG`) enhances the translation quality by 0.6 (en-it) and 0.2 (en-de) BLEU, with the improvement on en-it being statistically significant over the version without CTC compression. Moving it to previous layers (`4L PH AVG`, `2L PH AVG`) causes performance drops, suggesting that many layers are needed to extract useful phonetic information.

Then, we compare the different compression policies (rows 6-8): `AVG` outperforms (or matches) `WEIGHTED` and `SOFTMAX` on both languages. Indeed, the small weight these two methods assign to some vectors likely causes an information loss and prevents proper gradient propagation for the corresponding input elements.

Finally, we experiment with different CTC targets (rows 9-10), but both the phones without the position suffix (`8L PH W/O POS. AVG`) and the transcripts (`8L EN AVG`) lead to lower scores.

The different results between ASR and ST can be explained by the nature of the two tasks: extracting content knowledge is critical for ST but not for ASR, in which a compression can hide details that are not relevant to extrapolate meaning, but needed to generate precise transcripts. The RAM savings are higher in ST than in ASR as there are 3 more layers. On the 8th layer, they range from 11% to 23% for en-it, 16% to 22% for en-de. By moving the compression to previous layers, we can trade performance for RAM requirements, saving up to 50% of the memory.

We also tested whether we can use the saved RAM to add more layers and improve the translation quality. We added 3 encoder and 2 decoder layers: this (`8L PH AVG (14+6L)`) results in small gains (0.2 on en-it and 0.1 on en-de), but the additional memory required is also small (the RAM usage is still 10-16% lower than the Baseline). The improvements are statistically significant with respect to the models without compression (`8L PH`) on both language pairs. As such, we can conclude that the proposed CTC compression produces performance improvements and computational savings, leading to higher-quality and more efficient direct ST models.

3.4.5 Summary

After the work on knowledge transfer from MT (§3.3), we subsequently directed our efforts toward enhancing the quality and computational efficiency of direct ST models. This second strand of activities specifically focused on accounting for the different informativeness of the vectors representing the input sequence. Toward this goal, this section introduced a dynamic length reduction of the sequence representing the input audio in the encoder of direct ST models. Our experiments showed that our dynamic compression of the input improves the translation quality and reduces the memory footprint, allowing for training deeper models. In particular, the best approach consisted in averaging the vectors corresponding to the same phone prediction according to the CTC. The best model with such compression is able to outperform a strong baseline, which uses transcripts in a multi-task training, by 1.3 (en-it) and 1.5 (en-de) BLEU, reducing memory usage by 10-16%. These experiments demonstrated that accounting for the information variability in audio signals is a promising direction. In the next section, we exploit this CTC compression mechanism to propose a more complex architecture that avoids any fixed downsampling of the input.

3.5 Speechformer

Based on the promising results presented in the previous section, we introduce the first architecture for ST that does not reduce the length of the input audio sequence by means of an initial fixed downsampling. In this way, we prevent the loss of potentially useful linguistic information. Our architecture, which we named Speechformer, exploits a novel attention layer with reduced memory requirements (§3.5.1) and aggregates information only at a higher encoding level according to more informed linguistic criteria. Such compression reduces the redundancy of the more informative but longer resulting sequences, and enables the application of the traditional attention (§3.5.2). As we will see in §3.5.4, experiments on three language pairs (en→de/es/nl) show the efficacy of our solution, with gains of up to 0.8 BLEU on the standard MuST-C corpus and of up to 4.0 BLEU in a low resource scenario. We conclude the section with a manual analysis unveiling the main reasons for the gains, ascribable to a better audio and prosody understanding (§3.5.5).

3.5.1 ConvAttention layer

State-of-the-art ST models employ convolutional neural networks to sample the feature sequence to a lower dimension (typically by a factor of 4), enabling the use of Transformer layers otherwise impossible given their memory consumption. As described in §3.2.3, many works proposed methods to reduce the quadratic complexity of the product between the attention matrix, but they are hard to directly apply to ST. In the case of Linformer (Wang et al., 2020c), the main problem is the gradient inconsistency among the different GPUs as different subset of parameters of the introduced linear projection are updated.

To avoid this issue, we propose the adoption of ConvAttention (Figure

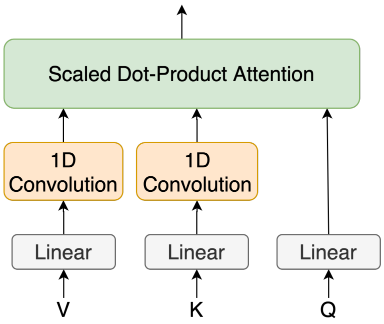


Figure 3.4: Attention mechanism with the proposed convolutional compression of K and V.

3.4), in which the linear projections of the Linformer architecture are substituted, both in K and V, with a single 1D convolutional layer. Hence, the length of the sequences used in the scaled dot-product attention depends on the stride of the convolution, a hyper-parameter we named *compression factor* (χ), which controls the memory complexity of the ConvAttention. Namely, being n the temporal dimension of K and V, the convolution output length is $\frac{n}{\chi}$ and the complexity of the ConvAttention is $O((\frac{n}{\chi})^2)$, i.e. a $\frac{1}{\chi^2}$ factor lower than a vanilla Transformer self-attention. For instance, setting χ to 4 leads to the same memory consumption as standard ST models with an initial ×4 subsampling (i.e. with two initial convolutional layers with stride 2).

Notice that the output sequence length is still equal to the input sequence length as it depends on the length of Q that is not modified.

3.5.2 Encoder Architecture

The introduction of ConvAttention layers allows us to avoid suboptimal fixed compressions that disregard the variability over time in the amount of audio information. However, since an encoder consisting only of ConvAttention layers does not compress the length of the original input sequence, the decoder would be fed with long and redundant sequences that are difficult

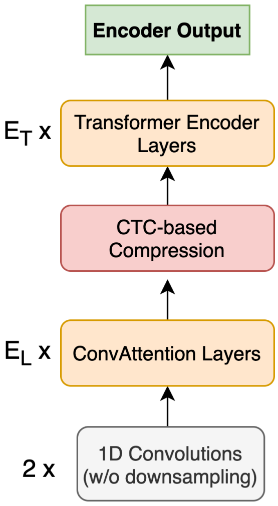


Figure 3.5: Speechformer architecture with E_L ConvAttention Layers and E_T Transformer Encoder Layers.

to attend, leading to potential performance degradation.

To overcome this problem, we apply the CTC compression described in the previous section, using as reference the sequence of subwords representing the transcript of the input utterance. After this operation, the sequence is reduced to a representation dimensionally closer to its textual content, which can be processed by the original attention mechanism without the need of approximations.

Speechformer (see Figure 3.5), is composed of E_L ConvAttention layers up to a CTC compression layer, after which there are E_T Transformer encoder layers. The E_L ConvAttention layers are meant to learn the linguistic content of the input audio while the E_T Transformer encoder layers are in charge of learning higher-level semantic representations, i.e. the encoder outputs, which the decoder has to convert into a text in the target language. We also maintain the two 1D convolutional layers before the ConvAttention layers but without striding, so that no sub-sampling is applied to the input. We make this choice both to keep the number of parameters comparable to the existing architectures, and to let the model

kernel	16	16	8	8	4
χ	16	8	8	4	4
BLEU	19.7	20.6	20.5	**21.3**	20.2

Table 3.8: BLEU on MuST-C en-de dev set varying the compression factor χ and 1D convolutional kernel size. The scores are obtained without label smoothing.

learn a better representation of the input before feeding it to the attention mechanism.

3.5.3 Experimental Settings

Data and Evaluation

We experiment on three languages of MuST-C: English-German (en-de), English-Spanish (en-es), and English-Dutch (en-nl). Differently from the previous sections, we extract 80 features from the input audio. The text is segmented in sub-word units with transcript and target Sentencepiece (Kudo and Richardson, 2018) unigram language models (Kudo, 2018) with size 5,000 and 8,000 respectively.

We evaluate the systems on the MuST-C test sets with SacreBLEU[9].

Architectures and Training Details

All our models are composed of 12 encoder layers and 6 decoder layers with 8 attention heads, 512 features for the attention layers, and 2,048 hidden units in the feed-forward layers. They are trained using label-smoothed cross entropy with the auxiliary CTC loss and Adam optimizer.

Following (Wang et al., 2020c), we share the convolution parameters of the ConvAttention layers both among K and V and among the attention heads. We select the compression factor and the 1D convolution kernel size with a set of preliminary experiments on the en-de validation set. The

[9] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0`

compression factor (χ) is chosen among 4, 8, and 16, since 4 is the minimum value that avoids out-of-memory issues. The kernel size is set either equal to or twice as the value of χ. Table 3.8 shows that the combination of a compression factor of 4 and a kernel size of 8 leads to better performance compared to the other combinations. Consequently, we use this setting in all our experiments.

We initialize the ConvAttention weights of Speechformer with those of a pre-trained ST model having only ConvAttention layers in the encoder, since, in the initial random state, the CTC-based compression might not properly reduce the input sequence, leading to out-of-memory issues in the following Transformer encoder layers.

The CTC is computed at the 8th encoder layer and its role is to predict the source transcription (lowercased and without punctuation), as in (Liu et al., 2020). The learning rate is set to 1e-3 with an inverse square-root scheduler and 10,000 warm-up updates. Mini-batches contain up to 5,000 tokens, and we update gradients every 16 mini-batches. We apply SpecAugment and utterance-level cepstral mean and variance normalization. We filter out samples with duration exceeding 30s. We average 7 checkpoints around the best on the validation loss. Trainings were performed with 4 K80 GPUs.

3.5.4 Results

After the comparison of the Speechformer architecture with the baselines, the section proceeds with an investigation of the behavior varying the amount of training data, and an analysis of the computational cost at inference time.

We compare the Speechformer architecture to a strong Baseline represented by a Transformer-based model with initial fixed sub-sampling (Wang et al., 2020b) and its Baseline+compression variant that includes the

Model	**en-de**				**en-es**				**en-nl**				**Inference**
	dev		tst-COMMON		dev		tst-COMMON		dev		tst-COMMON		**Time**
Inaguma et al. (2020)	-		22.9		-		28.0		-		27.4		-
Wang et al. (2020b)	-		22.7		-		27.2		-		27.3		-
Our Baseline	22.5		22.8		31.2		27.9		24.2		27.2		1.0x
+ compression	22.3	-0.2	22.8	+0.0	31.1	-0.1	27.9	+0.0	24.2	+0.0	27.0	-0.2	0.9x
Plain ConvAttention	23.1*	+0.6	23.2	+0.4	31.5	+0.3	27.7	-0.2	24.8*	+0.6	26.9	-0.3	1.8x
Speechformer	**23.3***	**+0.8**	**23.6***	**+0.8**	**31.8***	**+0.6**	**28.5***	**+0.6**	**24.9***	**+0.7**	**27.7***	**+0.5**	1.3x

Table 3.9: BLEU score (average over 3 runs) on English→Dutch (en-nl), English→German (en-de), and English→Spanish (en-es) of MuST-C tst-COMMON and dev set. The * symbol indicates statistically significant improvements over the baseline. Statistical significance is computed with a t-test (Student, 1908), whose null hypothesis is that the mean of the considered experiment is not higher than the mean of the baseline. We consider the result statistically significant if we can reject the null hypothesis with 95% confidence.

average CTC compression strategy (see §3.4). We choose to also develop the second baseline to make the comparison with Speechformer fair, since they both use the CTC compression strategy. Table 3.9 reports the results. For each experiment, we report the average over 3 runs to ensure that performance differences do not depend on the fluctuations of particularly good or bad runs.

First, it can be noticed that our Baseline is in line with state-of-the-art architectures trained only on MuST-C (Wang et al., 2020b; Inaguma et al., 2020). Second, the addition of CTC compression to the Baseline model does not bring benefits. This confirms the findings of the previous section, which showed that applying CTC compression using transcripts produces differences in scores that are not statistically significant. In these experiments, using phonemes did not bring significant improvements either, most likely due to the different experimental settings, so we do not include the results with them. Speechformer, instead, results in statistically significant improvements over the baseline in all language directions, with BLEU gains ranging from 0.5 (for en-nl) to 0.8 (for en-de). As the CTC compression is not helpful for the baseline, we also evaluate a model (*Plain ConvAttention*) whose encoder is a stack of ConvAttention layers, i.e.

without vanilla Transformer-encoder layers and any form of compression. The drop in performance with respect to Speechformer varies between 0.4 and 0.8 BLEU on all language pairs, supporting our hypothesis that a non-compressed encoder output is too redundant to be effectively attended by the decoder.

Low-Resource Settings. We suppose that the higher gains on en-de may be related to the size of the training data. Indeed, the en-de section of MuST-C used for training is the smallest one, containing 20% fewer data than the en-es section and 10% less than the en-nl one. Thus, we study the performance of Speechformer in different data conditions by progressively reducing the amount of training data. For this analysis, we select the en-es section of MuST-C as it contains the highest number of hours (478h) among the three languages, and we experiment with three subsets, respectively containing 385h (corresponding to the amount of training data for en-de), 200h, and 100h (which can be considered a limited quantity given that the number of hours is respectively less than half and one fourth of the available data). Figure 3.6 shows that the gains obtained by Speechformer over the Baseline do not vary significantly between 385h and 478h (0.5 vs 0.6 BLEU). We can then conclude that the gain variation between en-de and en-es does not depend on the smaller size of the en-de training set. However, in the low resource settings (200h and 100h), the gains obtained by the Speechformer are much larger, amounting to 1.1 BLEU with 200h and 4.0 BLEU with 100h. To validate the robustness of these results, we also experimented on the en-de language pair and obtained consistent results: Speechformer outperforms the Baseline by 1.5 BLEU (19.6 vs 18.1 BLEU) with 200h of training data and by 1.9 BLEU (9.7 vs 7.8 BLEU) with 100h of training data, achieving a considerable relative improvement of more than 24%. Although it brings consistent and significant gains in higher resource

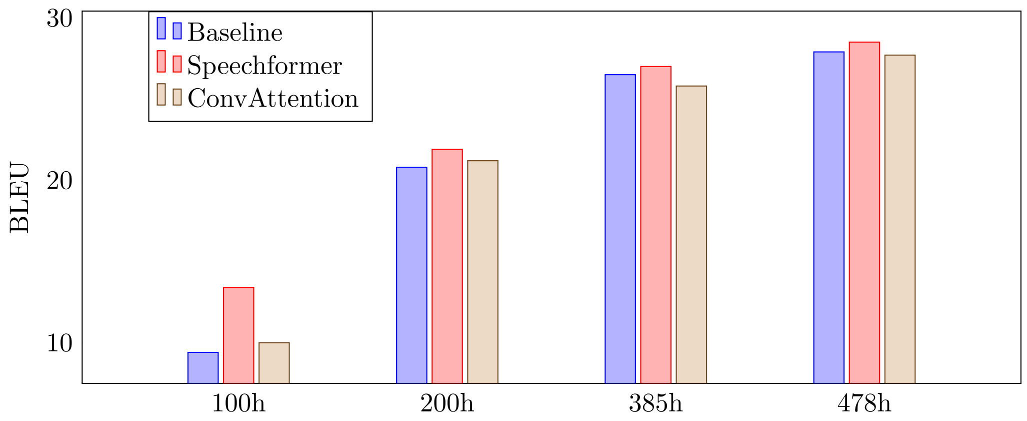


Figure 3.6: Architecture comparison varying the amount of en-es training data (478h, 385h, 200h, and 100h).

scenarios, these experiments show that Speechformer is particularly fruitful in low-resource settings.

Inference Time. The ConvAttention layers process the whole input sequences, which are 4 times larger than those elaborated by the baseline attention mechanism. Thereby, a slow-down at inference time is expected, especially for the *Plain ConvAttention*, whose encoder layers are all ConvAttention layers. The last column of Table 3.9 confirms that the *Plain ConvAttention* architecture is 1.8 times slower than the Baseline, i.e. the inference time is nearly twice. Speechformer is also slower than the Baseline, but the overhead amounts to only 30% instead of 80%. Moreover, it can be noticed that the size of the attention matrix – and therefore the corresponding computational cost – can be controlled in the Speechformer with the *compression factor* (χ) hyperparameter.

3.5.5 Analysis

Lastly, we inspected the Baseline and Speechformer outputs to better understand the reason behind the improvements brought by our architecture.

(a) Word ordering	
Audio	It was a way that parents could figure out which were the right public **schools** for their **kids**.
Reference	Es ging um eine Methode, mit der Eltern herausfinden können, welche die richtigen öffentlichen **Schulen** für ihre **Kinder** sind.
Baseline	Es war eine Möglichkeit, dass Eltern herausfinden konnten, welche für ihre **Kinder** die richtige öffentliche **Schule** war.
	*It was an opportunity for the parents to find out which were for their **children** the right public **schools**.*
Speechformer	Es war eine Methode, mit der Eltern herausfinden konnten, welche die richtigen öffentlichen **Schulen** für ihre **Kinder** waren.
	*It was a method with which the parents could find out which were the right public **schools** for their **children**.*
(b) Punctuation handling	
Audio	**So, sir, can you help me?** I need help.
Reference	**Also, mein Herr, können Sie mir helfen?** Ich brauche Hilfe.
Baseline	Es ist also möglich, mir zu helfen.
	So it is possible to help me.
Speechformer	**Also, können Sie mir helfen?** Ich habe keine Hilfe.
	So can you help me? *I have no help.*
(c) Audio misunderstanding	
Audio	You see Aluminum was the most valuable metal on the Planet, worth more than Gold and **Platinum**.
Reference	Aluminium war zu dieser Zeit das wertvollste Metall auf dem Planeten, wertvoller als Gold und **Platin**.
Baseline	Aluminium war die wertvollste Metallart auf dem Planeten, mehr als Gold und **Pflanzen**.
	*Aluminum was the most valuable type of metal on the planet, more than gold and **plants**.*
Speechformer	Aluminium war das wertvollste Metall auf dem Planeten, mehr als Gold und **Platin**.
	*Aluminum was the most valuable metal on the planet, more than gold and **platinum**.*
(d) Omission	
Audio	But the amazing thing about cities is they're worth so much more than **it costs** to build them.
Reference	Aber das Erstaunliche an Städten ist, dass sie so viel mehr wert sind, als **es kostet** sie zu bauen.
Baseline	Aber das Faszinierende an Städten ist, dass es viel mehr wert ist, als es zu bauen.
	But the fascinating thing about cities is that it's worth a lot more than building it.
Speechformer	Aber das Erstaunliche an Städten ist, dass sie viel mehr wert sind als sie **es kostet**, sie zu bauen.
	*But the amazing thing about cities is that they are worth a lot more than **it costs** to build them.*

Table 3.10: Examples of translation problems – *(a)*, *(b)*, *(c)* – and omissions – *(d)* – that Speechformer does not suffer from while Baseline does.

This qualitative analysis was conducted on a sample of 200 sentences of the en-de test set – the language direction showing the largest gap between the systems (+0.8) – by a professional linguist with C2 German level.

It emerged that Speechformer tends to have better word ordering, a typical problem arising when translating from an SVO language like English to an SOV language like German. Furthermore, Speechformer outputs display a better punctuation positioning – attributable to improved handling of pauses and prosody – and a reduction of the number of audio misunderstandings and omissions. Table 3.10 provides examples of the German translations generated by the Baseline and by Speechformer for four utterances of the MuST-C test set, selected to highlight the specific aspects that are better handled by our architecture.

Example *(a)* exhibits a wrong word ordering present in the Baseline output, i.e. it anticipates "für ihre Kinder" (*for their kids*) with respect to "die richtigen öffentlichen Schulen" (*the right public schools*). Speechformer, instead, translates the sentence in the correct order, making the translation easier to read and understand. Also, punctuation handling is improved by our model that, by leveraging the prosody present in the audio, is capable of detecting a question (i.e. *So can you help me?*) and translating it, as shown in example *(b)*. On the contrary, the Baseline does not capture these audio characteristics and does not translate the input in question form, besides omitting the last part of the reference sentence. The improved encoding of audio features by the Speechformer is also reflected in its superior understanding of audio content. This emerges from example *(c)*, where the word *Platinum* is correctly recognized and translated by our system, while the Baseline misunderstands and translates it in another word, "Pflanzen" (*plants*), with a completely different meaning. The better audio understanding of the Speechformer is present in example *(d)* as well. Indeed, the Baseline omits part of the original sentence (i.e. *it costs*), with a huge impact on the meaning of the resulting sentence, while Speechformer does not lose audio details and produces a complete translation. In this example, we can also notice that our system better handles pronominal references as it chooses *sie*, which follows the grammatical gender and number of *Staedten* (i.e. plural feminine), while the Baseline uses *es*, which wrongly agrees with *das Faszinierende* (i.e. singular neuter).

All in all, the manual inspection of the outputs indicates that Speechformer better captures the information present in the audio, and this improvement is reflected in more accurate outputs.

3.5.6 Summary

In the wake of previous results (§3.4) showing the benefits of a content-informed compression, we presented Speechformer: the first ST Transformer-based model able to encode the whole raw audio features without any suboptimal initial subsampling typical of current state-of-the-art models. Our solution is made possible by the introduction of a modified attention mechanism – the ConvAttention – that reduces the memory complexity to $O((\frac{n}{\chi})^2)$. The redundant sequences produced by the plain application of ConvAttention layers are compressed with a CTC-based strategy to obtain a compact, yet informative representation that vanilla Transformer encoder layers can process. Our experiments on three language pairs have shown that Speechformer significantly outperforms a state-of-the-art ST model by 0.5-0.8 BLEU, reaching a peak of +4 BLEU points in a low-resource scenario. However, we have also noted that Speechformer requires a pre-training of the first part of the encoder, which causes an overhead in terms of training costs. The following section is devoted to addressing this issue, as well as integrating and comparing our solutions with the recent Conformer model.

3.6 Competitive ST without Pre-training

Our goal is not only to increase the quality of the ST systems, but also to limit their training costs. Toward this goal, we first propose solutions to avoid the encoder pre-training for the Speechformer and we integrate it with the Conformer architecture – recently introduced with compelling results in ST (Inaguma et al., 2021; Vyas et al., 2021) – creating the Speechformer Hybrid model (§3.6.1). Second, we explore data selection mechanisms to increase model quality and reduce training time (§3.6.2).

Our experiments (§3.6.4) show that avoiding any additional pre-training or transfer learning from ASR does not degrade the performance in a Conformer architecture with CTC compression on the MuST-C en-de section. Scaling to high-resource data conditions, we notice that the gap between an ASR pre-trained system and a system trained from scratch is closed only after a fine-tuning on in-domain data. Moreover, with the addition of a simple data filtering method, we achieve the new state-of-the-art score of 26.7 BLEU for a direct ST model that does not exploit external (audio or textual) resources on the popular MuST-C en-de tst-COMMON benchmark.

3.6.1 Speechformer Hybrid

In light of the superiority of Conformer over ST Transformer models, we create a composite architecture made of a first stack of 8 Speechformer layers and a second stack of 4 Conformer layers. Hereinafter, we refer to this architecture as Speechformer Hybrid. As a side note, we also experimented with replacing the ReLU activation functions in the decoder of our Conformer model with the squared ReLU, in light of the recent findings on language models (So et al., 2021) showing accelerated model convergence, decreased training time, and improved performance. Unfortunately, these benefits were not observed in our experiments, as the introduction of the squared ReLU caused a small performance drop (-0.2 BLEU) and did not improve the convergence speed of the model. So, we do not consider this change in the rest of the thesis.

As in the Speechformer architecture, the encoder starts with two 1D convolutions that do not perform any downsampling. Indeed, the modified self-attention mechanism (ConvAttention) reduces memory requirements and the length of the input sequence is shrunk only on top of 8 ConvAttention layers by means of the CTC-compression mechanism before feeding the sequence to 4 Conformer layers. However, in a randomly initialized state, the CTC compression may actually not reduce the input sequence (or only slightly), leading to OOM errors caused by the quadratic memory complexity with respect to the sequence length of the Conformer layers. This issue can be prevented by initializing the encoder layers up to the CTC-compression module with a pre-trained model whose encoder is made only of ConvAttention layers. However, this solution contrasts with our goal of reducing the computational cost by avoiding any pre-training. For this reason, we introduce two methods that ensure a minimal compression factor of the input sequence after the CTC-compression:

- **Max Output Length**: if the sequence produced by the CTC compression is longer than a threshold (a hyperparameter that we set to 1/4 of the maximum input sequence length[10]), we merge (averaging them) an equal number of consecutive vectors so that the final length of the sequence is inferior to the defined threshold. For instance, if the maximum input sequence length is 4,000 vetcirs, we set the threshold to 1,000. in this case, if a sample results in a sequence of length 2,346 after the CTC compression, we merge the first 3 vectors, then the vectors from the 4th to the 6th, and so on. We use 3 because it is the minimum compression factor that satisfies the length requirement.[11]

- **Fixed compression**: for a given number of epochs n_E (a hyperparameter) the CTC compression is disabled and replaced by a fixed compression that averages 4 consecutive vectors. In this way, we directly control the length of the sequence after the compression, resembling the fixed compression performed by the initial 1D convolutional layers of Transformer and Conformer ST models.

[10] This ensures that the resulting sequences are not longer than the maximum length obtained by the Transformer and Conformer architectures after the two 1D convolutions.

[11] A compression factor 2 would result in a sequence of length 1,173 – higher than the 1,000 threshold – while 3 produces a sequence of length 782.

3.6.2 Data Filtering

Easy methods to improve the quality of ST systems – and deep neural networks in general – consist in providing them with *more* data or *better* data. The first approach comes at the cost of longer training time and higher computational requirements. This makes the second approach more appealing and in line with the overall goal and spirit of this work. We hence focus on the definition of an efficient filtering strategy that improves the quality of our training data (and consequently of our models) without additional computational costs.

We start from the observation that ST models estimate the probability of an output text given an input audio $p(Y|X)$, and a good ST model assigns a low probability to erroneous samples, which are outliers of the $p(Y|X)$ distribution. Although training an ST model only to filter the training data would be extremely computationally expensive, we decided to adopt this method as an upper bound for comparison with easier and feasible strategies. In particular, for each sample in the training set, we computed the negative log-likelihood[12] (NLL) with a strong ST model trained on all the data available for the IWSLT 2022 competition (see §3.6.3) as a proxy of the probability of the sample. A high NLL means that a sample is unlikely, while a NLL close to 0 means that the sample has a very high probability. Based on this, we can filter all the samples above a threshold to remove the least probable ones.

To set the threshold, we draw a histogram on all the training sets (see Figure 3.7) that leads to the following considerations: *i)* each dataset has a different distribution, making it difficult to define a threshold valid for all of them, and *ii)* MuST-C has the highest NLL, meaning that it is more complex to fit for the model.

[12] The negative log-likelihood is defined as $-log(p(Y|X))$.

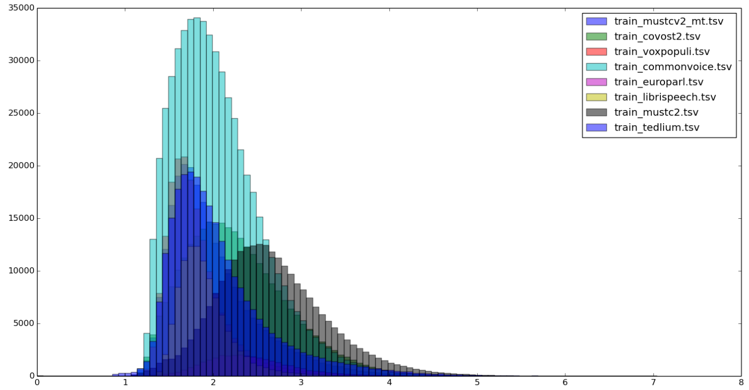


Figure 3.7: Histogram of the negative log-likelihood (NLL) of the samples for all the training set of the competition. The ST model used to estimate the NLL has been trained on all the data and was scoring 29.6 BLEU on MuST-C.

Through the approach described above, we selected the data of MuST-C with a NLL greater than 4.0. Upon a manual inspection of a sample of these selected data (5-10% of the total), we noticed that two main categories were present: *i)* bad source/target text alignments[13] (e.g. two sentences in the target translation are paired with only one in the transcript or vice versa), and *ii)* free (non-literal) translations. Instead, no cases of bad audio-transcript alignments were found (this was only a non-exhaustive manual inspection, though), meaning that this problem is likely less widespread and impactful than the textual alignment errors in the corpus.

These considerations motivated us to search for a feasible strategy to filter out the bad source/target text alignments. We first considered a simple method that discards samples with a too high or too low ratio between the target translation length (in characters) and the duration of the source audio.[14] The corresponding histogram on the training data can be found in Figure 3.8. Looking at the plots, it emerges that this ratio is

[13] In the MuST-C corpus, the alignments between transcripts and translations of the training set are automatically produced, hence misalignments and textual differences can be present.

[14] In practice, we compute the number of characters divided by the number of 10ms audio frames.

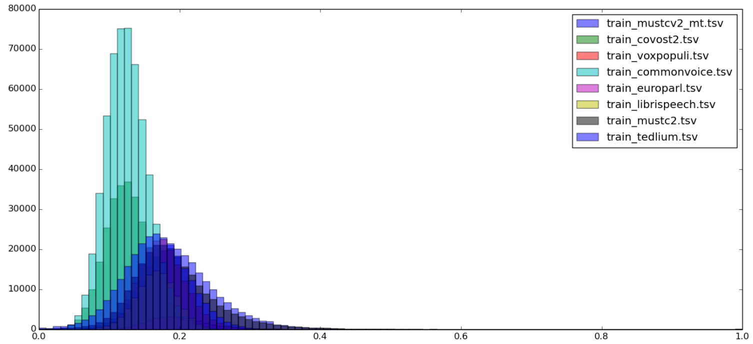


Figure 3.8: Histogram of the ratio between the number of target translation characters and 10ms audio frames for all the training set of the competition.

strongly dataset-dependent, likely due to the high variability in speaking rate for different domains and conditions, thus making it hard to set good thresholds. For this reason, also supported by the finding of our manual inspection on the good quality of audio-text alignments discussed above, we turn to examine the ratio between the target translation length and the *source transcript length*.[15] Figure 3.9 shows its histogram: in this case, the behavior is consistent on all datasets, making it easy to determine good values for the minimum and maximum acceptable ratio (we set them to 0.8 and 1.6).

3.6.3 Experimental Settings

Data and Evaluation

We perform a preliminary study on the English-German (en-de) section of MuST-C v2 and then we scale to the high-resource data condition to verify the preliminary findings. In high-resource settings, we include in the training set the ASR and ST datasets allowed for the offline tasks of

[15] We used normalized transcript without punctuation, so the length of the target translation is on average 1.2X that of the source transcript.

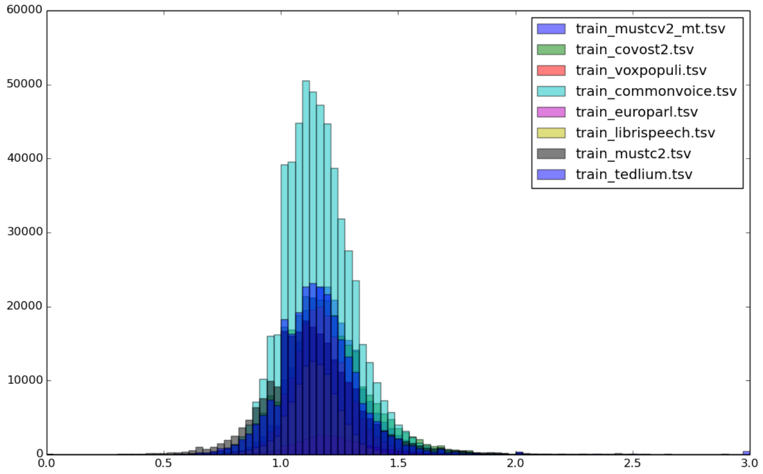


Figure 3.9: Histogram of the ratio between the number of characters in the target translation and the source punctuation-free transcript for all the training set of the competition.

IWSLT.[16] The ASR data consist in *(speech, transcript)* pairs that, in our case, are in English. The ST data consist in *(speech, transcript, translation)* triplets from a source language (here English) to a target language (here German). The ASR data we used are: LibriSpeech (Panayotov et al., 2015), TEDLIUM version 3 (Hernandez et al., 2018), Voxpopuli (Wang et al., 2021a), and Mozilla Common Voice.[17] The ST data we used are: MuST-C, CoVoST version 2 (Wang et al., 2020a), and Europarl-ST (Iranzo-Sánchez et al., 2020). The ASR-native corpora were included in our ST training by applying sequence KD from an MT teacher. The MT teacher is the freely available pre-trained model by Tran et al. (2021) for WMT2021 that was trained on the corresponding WMT2021 dataset (Akhbardeh et al., 2021).

As in the previous section, we extract 80 features from the input audio. The vocabularies are built via SentencePiece models. In our preliminary experiments only on MuST-C, the number of merge operations was set to 8,000 for the German translations and 5,000 for the lowercase punctuation-

[16] `https://iwslt.org/2022/offline`

[17] `https://commonvoice.mozilla.org/en/datasets`

free English transcripts. In the experiments on high-resource data condition, we doubled these values.

We evaluate translation quality with SacreBLEU[18] on the en-de section of MuST-C v1 and v2.

Architectures and Training Details

All the architectures (Transformer, Speechformer, Speechformer Hybrid, and Conformer) consist in 12 encoder layers and 6 decoder layers, 512 features for the attention layers and 2,048 hidden units in the feed-forward layers. We used 0.1 dropout for the feed-forward layer and attention layer. For Conformer convolutional layers we also apply 0.1 dropout and we set the kernel size to 31 for the point- and depth-wise convolutions.

We trained with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.98$). The learning rate was set to increase linearly from 0 to $2e-3$ for the first 25,000 warm-up steps and then to decay with an inverse square root policy. Differently, it was kept constant for model fine-tuning, with a value of $1e-3$. We normalize the audio features before passing them to our models with Cepstral Mean and Variance Normalization at utterance level.

Trainings were performed on 4 A100 GPUs. We set the maximum number of tokens to 40k per mini-batch and 2 as update frequency for the Conformer with CTC-compression. The other models were trained with 20k tokens per mini-batch and 4 as update frequency. We trained each model for 100,000 updates and averaged the last 7 checkpoints.

3.6.4 Results

We begin this section with a comparison of the architectures both with and without pre-training. After determining that with the best trade-off

[18] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0`

Model	**w pretrain**	**w/o pretrain**
Transformer	23.6	23.6
Speechformer	24.5	24.3
Conformer	24.8	24.8
+ CTC compr.	25.6	**25.5**
Speechformer Hybrid	**25.7**	24.9

Table 3.11: SacreBLEU on the *tst-COMMON* set of MuST-C v1 en-de.

between translation quality and computational efficiency, we compare the data filtering strategies described in §3.6.2, and validate our findings in high-resource conditions.

As a first step, we compare different architectures proposed for ST: ST Transformer, Conformer, Speechformer, and Speechformer Hybrid. For Speechformer and Speechformer Hybrid, we choose the n_E parameter of the fixed compression method (see §3.6) among the values 6, 8, 10, and 12 according to the BLEU score on the dev set. The best score was achieved with $n_E = 10$ (24.16 BLEU), which was lower than the score obtained by the Max Output Length method (24.26 BLEU). As such, in Table 3.11 (*w/o pretrain* column) we report the results of Speechformer and Speechformer Hybrid with the Max Output Length method.

The results show that the Speechformer-based models do need pre-training to reach their best scores while Conformer and Transformer models achieve comparable translation quality avoiding the pre-training. Specifically, the Conformer architecture with CTC compression obtains the best score without pre-training (25.5 BLEU) and has a negligible gap from the best result with pre-training (25.7 of Speechformer Hybrid). We can hence confirm the statement that, at least for Conformer and Transformer, ASR pre-training can be avoided at barely no translation quality cost. Therefore, in the rest of this section we use the Conformer with CTC compression without pre-training unless noted otherwise. It is worth mentioning that the introduction of the CTC compression in the Conformer encoder does

Model	**BLEU**
Cascade (Bahar et al., 2021)	25.9
Tight Integrated Cascade (Bahar et al., 2021)	26.5
Without external data	
SATE (Xu et al., 2021)	25.2
BiKD (Inaguma et al., 2021)	25.3
With external data	
JT-ST (Tang et al., 2021)	26.8
Chimera (Han et al., 2021)	26.3
This work	
Conformer + CTC compr.	25.5
+ char-ratio filter.	26.7
+ NLL-based filter.	26.9

Table 3.12: SacreBLEU on the *tst-COMMON* set of MuST-C v1 en-de. Chimera uses additional speech and WMT14 (Bojar et al., 2014), while JT-ST uses only WMT14 as external resource.

not only increase translation quality; also, it reduces the RAM requirements and speeds up both the inference and training phases. Indeed, as the sequence length is significantly reduced in the last encoder layers and in the encoder-decoder attention, fewer computations are required and the mini-batch size – the number of samples processed in parallel – can be increased. Overall, this leads to save $\sim 35\%$ of the training and inference time. We leave for future work further investigations on how to effectively train the Speechformer Hybrid models and on the reasons for the drop without pre-training, as, with encoder pre-training, it achieves the best scores.

Data Filtering

In Table 3.12 we report the results of our simple filtering method based on the target/source character ratio and we compare it with the upper bound of the NLL-based filtering strategy as well as with previous works both under the same data condition and with additional external data. First, we can notice that our method leads to a 1.2 BLEU gain, and has a very small gap (0.2 BLEU) with respect to the upper bound exploiting a strong ST model for filtering. Second, our score (26.7 BLEU) is significantly higher than those reported by previous direct ST works in the same data condition and is on par or even outperforms those of models trained with the addition of external resources. Finally, we compare the results of our model with those of the best cascade models reported in the same data conditions (Bahar et al., 2021): the tightly-integrated cascade is close to our model (-0.2 BLEU), but ours also benefits from the data filtering technique we just discussed.

To sum up, we managed to define a training recipe that enables reaching state-of-the-art ST results on MuST-C en-de (26.7 BLEU) with a single training step and involves: *i)* the Conformer architecture, *ii)* an auxiliary CTC loss and CTC-compression in the 8th encoder layer, and *iii)* a simple yet effective filtering strategy based on the ratio between source and target number of characters. In the following, we discuss the application of this procedure in high-resource data conditions.

High Resource Conditions

In addition to training our models in high-resource data conditions, we also investigate whether fine-tuning on in-domain data brings advantages or not. The results are reported in Table 3.13. As we can notice, the Conformer with pre-training outperforms its version trained from scratch by 0.9 BLEU. However, when both the systems are fine-tuned on the in-domain data (rows II and IV), this difference becomes negligible (0.1 BLEU) meaning that the pre-training phase can be skipped in favor of a single fine-tuning step. This might also suggest that the learning rate scheduler and the hyperparameters we used – tuned on MuST-C – may be suboptimal when

	Model	BLEU
I.	Conformer	30.6
II.	+ in-domain fn	31.6
III.	Conformer_pretrain	31.5
IV.	+ in-domain fn	**31.7**

Table 3.13: BLEU on MuST-C v2 tst-COMMON for Conformer with pre-training (*Conformer_pretrain*) and without it (*Conformer*). We also report the scores after fine-tuning on in-domain data (*+ in-domain fn*).

a large amount of data is available. For lack of large computing resources and time reasons, we did not investigate this aspect, which we leave to future work.

3.6.5 Summary

This section concludes our efforts toward high-quality and efficient direct ST models. In pursuit of these two objectives, we not only integrated the methods outlined in the previous sections with the recent Conformer architecture, but we also studied both in controlled settings (MuST-C) and in high resource conditions if we can skip complex and long training procedures without compromising the translation quality. To this aim, we *i)* showed that ASR pre-training of the encoder can be avoided without a significant impact on the final system performance, *ii)* proposed a simple yet effective data filtering technique to enhance translation quality while reducing the training time. The effectiveness of our solutions is demonstrated by the result (26.7 BLEU) on MuST-C en-de of our Conformer model with CTC compression that is – to the best of our knowledge – the best score reported in literature without leveraging external data or knowledge (e.g. in the format of pre-trained models – Gállego et al. 2021; Zhao et al. 2022; Polák et al. 2022).

3.7 Conclusions

This chapter described the efforts carried out in this PhD toward obtaining competitive results without undergoing long and computationally expensive training procedures. To this aim, our first strand of activities focused on the knowledge transfer from the MT models, which can benefit from an ample availability of corpora. Specifically, we showed that distilling knowledge from MT is a good knowledge transfer technique, but a KD-independent fine-tuning is required to solve the side effect of learning undesirable behaviors of the MT teacher, among which the most critical one is the truncation of multi-sentential utterances.

Then, we dedicated to modeling the audio information in a way that accounts for the variability of the amount of information over time typical of speech signals. Toward this goal, we demonstrated that averaging the vectors corresponding to the same phone prediction according to a CTC module improves the translation quality and reduces the memory footprint as well as the training/inference time. Along the same research direction, we also removed the initial fixed subsampling of the audio sequence and the information loss it causes thanks to a more efficient attention mechanism, leaving the CTC content-based compression as the only downsampling method in the encoder. Lastly, we proved that the pre-training of the ST encoder with the weights of an ASR model can be avoided without a significant impact on the final system performance, and that a simple yet effective data filtering technique enhances translation quality while reducing the training time.

The results of the efforts of these three years, combined with the improvements coming from the research community, brought direct ST to the level of (or even above) cascade solutions. Indeed, we scored 26.7 BLEU on the popular MuST-C v1 en-de benchmark without introducing external data, 0.2 BLEU more than the best cascade solution published in the same data condition, to the best of our knowledge. In light of the high translation quality achieved by direct ST models, in the following chapters, we focus on more specific issues, yet essential for their adoption in real applications. Chapter 4 goes beyond the ideal condition in which, at test and inference time, the audio signal is already split into utterances containing well-formed sentences. Chapter 5 inspects the bias and potential harms to different groups of users, focusing on gender disparities. Chapter 6 investigates the employment of direct ST systems in the context of the augmented ST paradigm, in which the ST system supports the users by highlighting relevant information.

Chapter 4

Audio Segmentation

4.1 Introduction

The previous chapter described our efforts toward higher-quality direct ST models, which were trained and evaluated on corpora segmented at the sentence level. Indeed, existing corpora split continuous speech into utterances according to strong punctuation marks in the transcripts (which are known in advance), reflecting linguistic criteria related to sentence well-formedness. This "gold" segmentation is optimal, as it allows ST systems to potentially generate correct outputs even for languages with different syntax and word order (e.g., subject-verb-object vs subject-object-verb). However, audio transcripts are not known in advance at inference time and other segmentation techniques have to be applied. The traditional approach consists in adopting a Voice Activity Detection (VAD) tool to break the audio on speaker silences (Sohn et al., 1999), considered as a *proxy* of clause boundaries. As such, the way the audio is split into segments is considerably different at training and inference time: this causes a mismatch between the data the models are trained on and the data they have to process at inference time, which severely harms the translation quality (Sinclair et al., 2014).

Both cascade and direct ST systems can be significantly affected by this mismatch between training and test data. In cascade systems, though, the impact of a syntax-unaware segmentation can be limited by means of dedicated components that re-segment the ASR transcripts, so as to feed the MT component with well-formed sentences (Matusov et al., 2006; Oda et al., 2014; Cho et al., 2017). The absence of intermediate transcripts makes this solution unfeasible for direct systems, whose performance is therefore highly sensitive to suboptimal audio segmentation.

In light of these considerations, we approached the problem from two different perspectives. On one side, we proposed methods to create ST models that are robust to automatically segmented audio and limit the quality drop from optimally segmented data (§4.3). This was done either by fine-tuning them on artificial data (§4.3.1) or by providing the previous segment as contextual information (§4.3.2). On the other side, we analyzed in depth the strengths and weaknesses of different audio segmentation methods in the context of direct ST. Based on the resulting observations, we introduced improved hybrid methods that can also be applied to streaming audio (§4.4). At last, we combined the approaches, studied how they interact, and compared them with newly proposed solutions on state-of-the-art Conformer models (§4.5).

Our contributions can be summarized as follows: *i)* we build models robust to automatic segmentation by re-segmenting training corpora with random utterance boundaries and fine-tuning the models on them; *ii)* we introduce the first context-aware direct ST models, showing that they outperform a strong base model and the fine-tuning on different VAD segmentations of an English-German test set by up to 4.25 BLEU points; *iii)* we propose enhanced hybrid solutions (based on both utterance length and audio properties) that reduce by at least 30% the gap between the traditional VAD-based approach and optimal manual segmentation; *iv)* we combine the two approaches, apply them to state-of-the-art Conformer models, and compare them with recent alternative solutions.

4.2 Related Works

In this section, we first provide an overview of the audio segmentation methods available at the time of writing this thesis (§4.2.1), which also covers methods posterior to our work on the topic presented in §4.4. Then, we discuss the works that study the integration of surrounding audio segments as contextual information in ST (§4.2.2), which are all posterior to our proposal in §4.3.

4.2.1 Audio Segmentation

Audio segmentation has been tackled with 4 main categories of approaches: *i)* VAD systems, *ii)* fixed-length methods, *iii)* hybrid solutions (considering both audio length and pauses), and *iv)* ASR-based models.

VAD systems. VAD tools are classifiers that determine whether a given audio frame contains speech or not. Based on this, a VAD-based segmentation considers a sequence of consecutive speech frames as a segment, filtering out non-speech frames. In the context of ST, the most widely used open-source VAD tools are: LIUM (Meignier and Merlin, 2010) and WebRTC's VAD.[1] For instance, to date all the IWSLT offline ST evaluation campaigns (Niehues et al., 2018, 2019; Ansari et al., 2020; Anastasopoulos et al., 2021, 2022) have released a default audio segmentation obtained with LIUM, although participants are free to use their own segmentation technique. Although easy and efficient, such methods are known to cause unpredictable, often large, performance drops (Sinclair et al., 2014).

[1] http://webrtc.org/. We use the Python interface http://github.com/wiseman/py-webrtcvad.

Fixed length. A simple approach is splitting the audio at a predefined fixed length (Sinclair et al., 2014), without considering the content. In contrast with VAD, this naive method has the benefit of ensuring that the resulting segments are neither too long nor too short, which are typically hard conditions for ST systems. However, the split points are likely to break sentences in critical positions, such as between a subject and a verb or even in the middle of a word. Unlike in the VAD solution, non-speech frames are not filtered from the input audio, which is entirely passed to the ST system.

Hybrid methods. The method described in (Potapczyk and Przybysz, 2020) takes into account both audio content (silences) and target segment length (i.e., the desired length of the generated segments) to split the audio. It recursively divides the audio segments on the longest silence, until either there are no more silences in a segment, or the segment itself is shorter than a threshold. It is important to notice that the silences are detected with a manual operation, making the approach hard to reproduce and not scalable. To compare with this method, we replicate the logic, but we rely on WebRTC to automatically identify silences. For this reason, our results might be slightly different from the original ones, but the segmentation is automatic and easy to reproduce. Another major problem of this method is that it requires the full audio to be available for splitting it. So it is not applicable to audio streams and online use cases.

ASR models. Bahar et al. (2020) exploited an external hybrid ASR model to segment the audio (showing a 10% BLEU gain compared to its VAD-based counterpart). This solution, however, formally makes direct ST closer to a cascade architecture, losing the advantage of the reduced latency of direct systems. Recently, Tsiamas et al. (2022b) presented a novel

Supervised Hybrid Audio Segmentation (SHAS) with excellent results in limiting the translation quality drop. SHAS adopts a probabilistic version of the *divide and conquer* algorithm by Potapczyk and Przybysz (2020) that progressively splits the audio at the frame with the highest probability of being a splitting point until all segments are below a specified length. The probability of being a splitting point is estimated by a classifier fed with audio representations generated by wav2vec 2.0 (Baevski et al., 2020) and trained to approximate the manual segmentation of the existing corpora, i.e. to emit 1 for frames representing splitting points and 0 otherwise. Since this approach involves a prediction with neural models of considerable size, its superiority over the VAD-based ones comes with a significant computational cost and overhead. In addition, SHAS is not applicable to audio streams, as it requires the full audio to be available before start splitting. In the context of offline ST, however, these limitations do not represent a significant issue.

4.2.2 Contextual ST

The idea of exploiting contextual information from previous sentences to improve translation quality has been introduced when MT was still performed with phrase-based approaches (Hardmeier et al., 2012; Xiong and Zhang, 2013) and it has been successfully applied in NMT (Wang et al., 2017; Zhang et al., 2018; Bawden et al., 2018; Kim et al., 2019). In light of this and following our work on the topic, several papers tested its effectiveness in providing contextual information in ST as well. Zhang et al. (2021) concatenated the previous audio sentences as input to the encoder, and provided the generated translations of these sentences as previous output tokens to the decoder. They showed that this approach provides consistent gains over all 8 language pairs of MuST-C, and their best results are obtained by feeding the previous 2 segments with the one to be translated. Similarly, Martínez De Morentin Cardoner (2022) investigated the concatenation of both the previous and the following segment with the current utterance, showing that both can contribute to small quality improvements, with the previous one being the more important. In our work, we do not explore solutions based on the concatenation of the context with the current input (Agrawal et al., 2018), or the combination of encoded representations of the two (Voita et al., 2018), for two reasons: *i)* they are highly inefficient (because of the quadratic complexity of the self-attention), and *ii)* previous findings in NMT (Kim et al., 2019) demonstrate the superiority of methods exploiting a dedicated attention. Closer to our solution, (Bang et al., 2022) processed the previously generated text with BERT (Devlin et al., 2019), and provided this encoder representation as context to the decoder, showing that this approach helps the translation of unclearly spoken utterances. None of the above papers, though, explore the effectiveness of context-based solutions to recover from the information loss caused by a suboptimal automatic audio segmentation, as we will do in §4.3.

4.3 Model Robustness to Automatic Segmentation

In this section, we move our first step toward reducing the negative effects of the train/inference audio-segmentation mismatch by making our direct ST models more robust to automatically segmented data. In particular, we focus on the traditional and widespread approach in which the audio is segmented with VAD tools. As seen in §4.2.1, VAD systems determine whether a given short (usually 10-30 ms) audio segment actually contains speech. This information is used in the context of ST for two purposes: *i)* dividing the audio stream into segments containing uninterrupted speech; *ii)* filtering out audio segments containing other sounds. Since VAD is solely based on the alternation between human voice, silences and other sounds, the resulting splits might not correspond to well-formed sentences but to fragments of one or more sentences. The impact of feeding an ST model trained on optimally segmented data with suboptimal, not linguistically-motivated segmentations varies according to the characteristics of the VAD employed and its settings. Very aggressive settings reduce the generation of long (cross-sentential) segments, which are difficult to handle by neural models that are typically very sensitive to input length. On the downside, they produce short (sub-sentential) segments that might not provide enough context for proper translation.

In light of this, the contribution of this section is the definition of architectural solutions and training recipes that increase the ability of direct ST models in translating cross- and sub-sentential utterances. To this aim, we first generate an artificial dataset by randomly re-segmenting clean (i.e., sentence-based) ST data. Then, we train ST systems on this new dataset with the aim of reducing the distributional shift (or mismatch) with the data fed at inference time. In particular, we experiment with two approaches: *i)* fine-tuning on the new dataset (§4.3.1); *ii)* improving our direct ST model with the capability to look back and attend to the preceding segment as contextual information (§4.3.2). Our experiments (§4.3.4) show that the context-based solution effectively handles the segmentation of different VAD systems and configurations, reducing the drop in translation quality caused by segmentation mismatches in the training and test data by up to 55%.

4.3.1 Training on Automatically Segmented Data

As a first solution to "robustify" ST models with respect to segmentation mismatches between the utterances used for training and those automatically segmented at inference time, we propose to fine-tune ST models on an automatic re-segmentation of the training set. The method consists in choosing a random word in the transcript of each sample, and using it as the sentence boundary instead of the linguistically-motivated (sentence-level) splits provided in the original data.

The re-segmentation starts by picking a random (with uniform distribution) *split word* for each sample in the original English transcripts. Each fragment spanning from a *split word* to the word before the next *split word* becomes a segment of the new training set. We extract the audio corresponding to each resulting transcript by leveraging word alignments computed with Gentle.[2] Then, we retrieve the corresponding translations using word alignments generated with fast_align (Dyer et al., 2013). In case of missing alignments (either with the audio or with the translation), the sample is discarded. The resulting training dataset contains 4K samples less than the original (225K vs 229K), while the validation set size is almost unchanged.

A manual check on a sample of the produced aligned segments revealed that about 96% of them are acceptable. The most frequently observed issue is that some translations contain 1-2 words more than the optimum, mostly due to the lack of some word alignments and word reordering. Although this caused no issue when fine-tuning the model, in the next section we will present a use case in which, instead, it represents a problem and how it can be easily addressed.

4.3.2 Contextualized Translation

A second approach consists in looking at the previous segment as contextual information to recover from information losses caused by suboptimal (sub-sentential) splits. As seen in §4.2.2, the idea of exploiting contextual information to improve translation is not new, as it has been successfully applied in MT. In our use case, we are interested only in modeling short-

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

range cross-segment dependencies to cope with the suboptimal breaks introduced by automatic segmentation. We hence consider as context only the segment immediately preceding the one to be translated, leaving out of our study hierarchical approaches modeling the whole document as context. Moreover, while in document-level NMT the best approach is to use the source side of the sentence(s) as contextual information, in the ST scenario it is not trivial to understand which side is the best. On the one hand, audio source avoids the error propagation and exposure bias introduced by using as context the translations generated at inference time. On the other, these problems are balanced by the easiness of extracting information from text rather than from audio (Di Gangi et al., 2020b). Here, we study both options.

To integrate context information into the model, we explore the two solutions that gave the best results for NMT (Kim et al., 2019). They respectively use sequential (Zhang et al., 2018) and parallel (Bawden et al., 2018) decoders. We also experimented with the integration of context information in the encoder (Zhang et al., 2018), but the trainings were either very unstable (when using audio as context) or ineffective, eventually leading to worse results.

Both the sequential and the parallel decoder use a multi-encoder approach, with an additional encoder dedicated to the context information. The context encoder is composed of Transformer encoder layers, but its input depends on the modality of the segment used as context, i.e., text or audio. When we use the generated translation as context, its tokens are converted into vectors with *word embeddings* (namely, we re-use the decoder embeddings), summed with *positional encoding* and then provided to the encoder Transformer layers. When we use the audio as context, the input audio features are first processed by the encoder of the base model and then passed to the context encoder (Di Gangi et al., 2020b).

The difference between the two methods (sequential and parallel) lies in the way this information is integrated into the decoder of base model, which is described below.

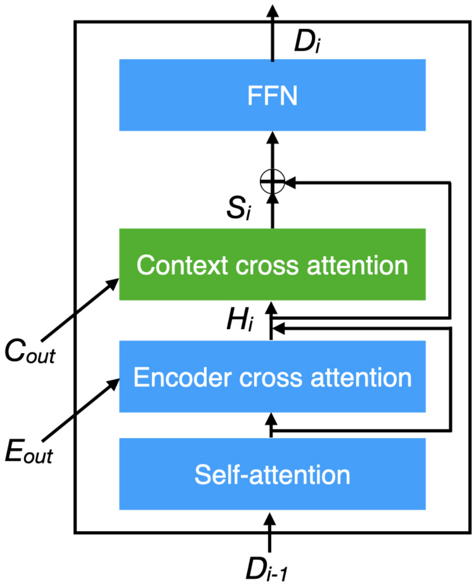


Figure 4.1: Sequential context integration.

Sequential (Figure 4.1). In each decoder Transformer layer, an additional multi-head cross-attention sublayer is introduced. It queries the output C_{out} of the context encoder using the output H_i of the i-th encoder cross-attention sub-layer. The result S_i of this operation is combined with H_i using a position-wise gating mechanism, before being fed to the feed-forward network FFN_i. Hence, the output of the i-th decoder layer D_i is:

$$\lambda_i = \sigma(W_{hi}H_i + W_{si}S_i) \tag{4.1}$$

$$D_i = \text{FFN}_i(\lambda_i H_i + (1 - \lambda_i)S_i) \tag{4.2}$$

Parallel (Figure 4.2). In each decoder Transformer layer, the output of the self-attention sublayer is used as query for both the encoder cross-attention

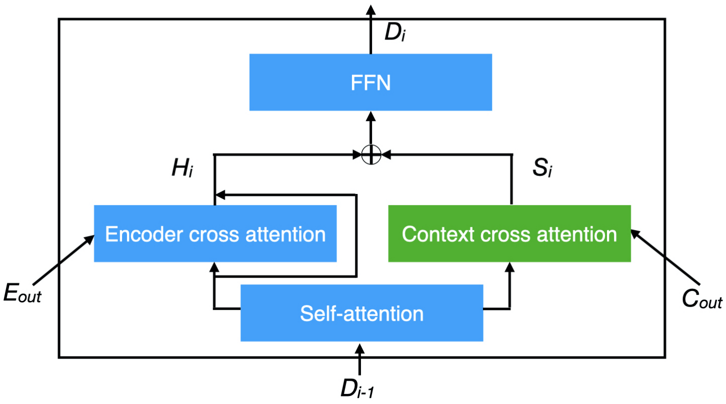


Figure 4.2: Parallel context integration.

and the context cross-attention defined in the same way as in the previous case. The outputs of these two sub-layers are then combined using the position-wise gating mechanism described in Eq.(4.2).

To avoid over-relying on the context, we add a regularization on the context gate. Our regularization is slightly different from the one proposed by Li et al. (2020): we always penalize the context information, so that the model will use it only when it is strictly needed. With the regularization factor, the resulting loss is:

$$\mathcal{L}' = \mathcal{L} + \alpha \sum_{i=0}^{N_d} (1 - \lambda_i) \tag{4.3}$$

To train the model, we rely on data segmented as described in the previous section (§4.3.1), in which the previous segment is considered as context. However, as mentioned above, the target side (translation) can contain extra words due to alignment and word-ordering issues. This leads to the presence of overlapping words between the context and the target references in 25% of the samples. In early experiments, this caused model instability at inference time because models learned to copy the final context words, up to producing nonsensical sequences of repeated tokens. We solved the issue by filtering out the overlapping words from the context.

4.3.3 Experimental settings

Data and Evaluation

We performed preliminary experiments on a baseline model (hereinafter *BASE_MUSTC*) trained the on English-German data drawn from the MuST-C corpus. Since models using the generated translations as context are affected by exposure bias, we wanted to test our solution also in more realistic conditions, with a stronger model trained in rich data conditions. This model (hereinafter *BASE_ALL*) was trained on all the data available for the IWSLT 2020 evaluation campaign.[3] Textual data were pre-processed with tokenization and punctuation normalization performed using Moses (Koehn et al., 2007), and were segmented with $8,000$ BPE merge rules. The audio was preprocessed as described in §3.3.2.

As we want our systems to be robust to different VAD outputs, we test our models on both LIUM and the WebRTC VAD.[4] For LIUM, we apply the configuration employed in the IWSLT 2020 campaign (Ansari et al., 2020). For WebRTC, we tested all the possible configurations, varying the *frame size* (allowed values are 10ms, 20ms and 30ms) and the *aggressiveness* (ranging from 0 to 3, extremes included). We discarded those producing either too long ($>$ 60s) or too many segments ($> 5,100$, i.e. twice the segments of the original sentence-based segmentation of the MuST-C test set). In this way, we ended up with three configurations, whose characteristics are described in Table 4.1.

Overall, the segments produced by WebRTC have much higher variance in their length (ranging from 0.40s to 58.62s) compared to LIUM (from 2.50s to 18.63s) and are significantly more ($> 3,500$ vs $2,725$). This can affect the final performance of neural ST models, for which handling very long/short segments is difficult. However, from a qualitative standpoint, a

[3] http://iwslt.org/doku.php?id=offline_speech_translation

[4] We use the open-source Python interface `https://github.com/wiseman/py-webrtcvad`.

System	Man.	LIUM	WebRTC		
Frame size			30ms	20ms	20ms
Aggress.			3	2	3
% filt. audio	14.66	0.00	11.27	9.53	15.58
Num. segm.	2,574	2,725	3,714	3,506	5,005
Max len. (s)	51.97	18.63	48.84	58.62	46.76
Min len. (s)	0.05	2.50	0.60	0.40	0.40

Table 4.1: Statistics for different segmentations of the MuST-C test set. "Man." refers to the original sentence-based segmentation.

manual inspection of 50 samples showed that the split times selected by LIUM are less accurate than those selected by WebRTC: while the former often splits fluent speech, the latter always selects positions in which the speaker is actually silent.

Evaluation is performed with BLEU[5] (Papineni et al., 2002) and TER (Snover et al., 2006).

Architecture and Training Details

Our architectures are analogous to the big models introduced in §3.3.2 unless stated otherwise. The number of context encoder layers N_c is set to 1, as (Zhang et al., 2018) shows that this leads to the best results. Since (Kim et al., 2019) has demonstrated that poorly regularized systems can lead to ambiguous results when integrating context, we used 0.2 dropout and *SpecAugment* to prevent this issue.

The *BASE_MUSTC* model has 8 encoder layers N_e and 6 decoder layers N_d. The *BASE_ALL* model, instead, has N_e set to 11 and N_d to 4. The training of this latter model involves a pre-training on the synthetic data, a fine-tuning on the data having ground-truth translations and a second fine-tuning using label-smoothed cross entropy instead of knowledge distillation,

[5] Computed with the `multi-bleu.pl` script.

as per §3.3.

All our models are optimized with label smoothed cross entropy using the Adam optimizer (Kingma and Ba, 2015) with a learning rate starting from $3 \cdot 10^{-4}$, increasing linearly up to $5 \cdot 10^{-4}$ in the first $5,000$ steps and then decaying with inverse square root policy. We train on 4 K80 GPUs using 8 sentences as mini-batch size and 16 as update frequency.

All the context-aware models are initialized with the corresponding baseline model trained on sentence-segmented data. We experimented with freezing all the pre-trained parameters as in (Zhang et al., 2018), but freezing the decoder weights turned out to be harmful. If freezed, decoder layers are not able to adapt to the new inputs (with different segmentation) and this slows down convergence and leads to worse results. We hence freeze only the encoder.

We choose the best model according to the loss on the validation set.

4.3.4 Results

We performed preliminary experiments with *BASE_MUSTC* (scoring 21.08 BLEU on the original MuST-C en-de test set) to compare the context integration techniques and select the most suitable one for ST. We then compared the fine-tuning with the context-aware models using the stronger baseline model *BASE_ALL* (scoring 27.55 BLEU on the original test set).

Context information and integration

Table 4.2 shows that all the tested approaches outperform the baseline on VAD-segmented data with a margin that ranges from 0.25 to 2.69 BLEU points. This indicates that the context is useful to mitigate the effect of VAD-based segmentation. On LIUM, our models achieve the highest score (*TGT PAR*, 20.01 BLEU) and the largest gain over the baseline; on

	LIUM	**WebRTC**		
		3, 30ms	**2, 20ms**	**3, 20ms**
BASE_MUSTC	17.32	17.82	17.75	16.31
SRC SEQ	19.08	18.81	18.00	17.42
SRC PAR	19.25	18.90	18.25	17.30
TGT SEQ	19.57	**19.21**	18.81	**17.60**
TGT PAR	**20.01**	18.98	**18.82**	17.32

Table 4.2: Evaluation results on the VAD-segmented test set. Notes: SRC=audio as context; TGT=generated translation as context; SEQ=sequential; PAR=parallel.

WebRTC the improvements are significant but smaller. We argue that the reason lies in the different characteristics of the two tools. The split positions selected by LIUM do not always correspond to actual pauses in the audio, which prevents the baseline model from disposing of all the information necessary for translation. This information, instead, is available to the context-aware models as they can access the previous segment. WebRTC, instead, produces very long/short segments, whose effect on context-aware models is limited: the contribution of adding the previous segment is low both in case of very long segments, as only the first part is influenced by it, and in case of very short ones, as having a short segment as context means adding little information. We also experimented with including manually-segmented data in the training set of the models, but it was not beneficial for any of them.

Looking at the context modality (text vs audio), we observe that supplying the previously generated translation (TGT*) yields higher BLEU scores than supplying its corresponding audio (SRC*) with both the integration types (*SEQ and *PAR). This suggests that the audio representation produced by current ST models is less suitable than text to extract useful content information to support translation. In light of these observations, we decided to proceed with *TGT SEQ* and *TGT PAR* in the following

	LIUM		**WebRTC**					
			AGG=3, FS=30ms		**AGG=2, FS=20ms**		**AGG=3, FS=20ms**	
	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)
BASE_ALL	19.66	76.57	22.07	67.08	21.98	66.83	19.59	72.62
FINE-TUNE	22.48	64.21	23.48	60.03	**23.40**	61.54	21.35	63.90
TGT SEQ	23.18	**58.60**	22.85	**58.49**	22.59	**59.79**	21.11	**60.51**
+ REG	23.88	58.81	**23.61**	58.57	23.15	60.36	21.88	60.97
TGT PAR	23.77	59.02	23.34	58.94	22.91	60.09	21.75	60.77
+ REG	**23.91**	58.95	23.51	58.64	**23.40**	59.95	**22.03**	60.83

Table 4.3: Comparison between base model, fine-tuning and context-aware models.

experiments with the stronger *BASE_ALL* model.

Context vs fine-tuning

To disentangle the benefits produced by the context and those due to the use of artificial training data, we compare the performance of the fine-tuning and the context-aware solutions.

The results in Table 4.3 show that: *i)* fine-tuning on the artificial data produces significant gains over *BASE_ALL* (respectively, 2.82 BLEU points on LIUM and from 1.41 to 1.76 on WebRTC), and *ii) TGT PAR* outperforms *TGT SEQ* on all datasets (by 0.32 to 0.64). *TGT PAR* without regularization is superior to the fine-tuning when the VAD splits very aggressively (21.75 vs 21.35 on WebRTC 3, 20ms) or in non-pause positions (23.77 vs 22.48 on LIUM). On the other VAD configurations, the results are close, but inferior to the fine-tuning. Our intuition is that this behavior is caused by the noise added by the context-attention when the context is not needed. This is confirmed by the results obtained when adding the context-gate regularization[6] presented in Eq. (4.3) (*TGT PAR+REG*

[6] The value of the hyperparameter α was chosen among 0.01, 0.02, 0.04 and 0.08: we set it to 0.04 as it provided the best loss on the validation set.

and *TGT SEQ+REG*). The regularization allows our best context-aware model (*TGT PAR+REG*) to outperform the fine-tuned model on 3 out of the 4 VAD configurations tested (in one case BLEU is on par) and improves both integration types. *TGT SEQ* benefits more from it, closing the gap with *TGT PAR*.

An in-depth analysis of the BLEU scores revealed that 1-,2-,3- and 4-gram BLEU scores are always significantly higher for the context-aware solutions than for the fine-tuning, even when the overall BLEU scores are close (or on par). This gap, indeed, is not reflected in the final score due to the brevity penalty, as the context-aware models produce shorter translations. This difference between context-aware models and fine-tuning is confirmed and evident if we consider the TER metric (the lower, the better). In this case, *TGT SEQ* obtains the best scores in every setting, but the results of all context-aware models are close and are 2 to 6 points better than those obtained with fine-tuning. Interestingly, the best result (23.91 BLEU) is obtained by exploiting the context in one of the worst segmentations for the base model (19.66 BLEU). This is coherent with the behavior observed in §4.3.4.

4.3.5 Analysis

A researcher with a background in linguistics and excellent English knowledge performed a manual analysis of the translations produced by the baseline and by our best context-aware model (*TGT PAR* + REG) on the LIUM-segmented test set. The goal was to check whether the gains are actually due to the use of contextual information and to understand how this information is exploited. We noticed three main issues solved by the context-aware approach. They are all related to the presence of subsentential fragments located at the beginning or the end of a segment. First, these fragments are often ignored by the baseline model. Being trained only on well-formed sentences from the clean MuST-C corpus, this model seems unable to handle segments reflecting truncated sentences and, instead of returning partial translations, it opts for ignoring part of the input audio. Second, the base model produces *hallucinations* (Lee et al., 2018) trying to translate a sub-sentential fragment into a well-formed target sentence. Our models, instead, produce the translation corresponding to the incomplete fragment. Third, the baseline model translates the sub-sentential fragment and the adjacent sentence in the same segment into one single output sentence, mixing them. In contrast, context-aware models translate them separately.

4.3.6 Summary

As a first approach to reducing the translation quality drop caused by automatic (suboptimal) audio segmentation performed at inference time, in this section we studied how to make ST models robust to VAD-segmented utterances. To this aim, we explored different approaches to integrate contextual information provided by the segment preceding the one to be translated. Our experiments show that a context-aware architecture, trained on artificial data generated with random segmentation, improves final translation quality. We also demonstrate that, compared to the best automatic segmentation (22.07 BLEU), context-aware models achieve results that are similar in the worst case (22.03) and significantly better in the best case (23.91). In this case, our context-based systems reduce by 55% the performance gap of the base model (19.66) with respect to optimal (i.e. sentence-level) manual segmentation (27.55). These results confirm the effectiveness of this approach in reducing the drop in translation quality caused by segmentation mismatches in the training and test data. As an alternative and complementary method, the next section directly addresses the problem of the mismatch between the way the audio is segmented in the training data and at inference time and tries to mitigate the mismatch by proposing a better audio segmentation technique.

4.4 Hybrid Audio Segmentation

As a complementary approach to build robust models, in this section we focus on the audio segmentation itself. Audio segmentation strategies typically aim to mimic the sentence-based segmentation observed in the training data to reduce the distributional shift between training and inference inputs. The importance of the audio segmentation has been demonstrated in the 2020 IWSLT evaluation campaign (Ansari et al., 2020), where the best direct ST system had a key feature in the segmentation algorithm (Potapczyk and Przybysz, 2020), improving by 3.81 BLEU points the score achieved when using the basic segmentation provided by the task organizers. Similarly, in the 2021 edition (Anastasopoulos et al., 2021) the participants that employed their own audio segmentation method outperformed nearly all the teams that utilized the provided basic segmentation. At last, in 2022, the basic segmentation was definitively abandoned (Anastasopoulos et al., 2022).

Our goal is the introduction of a segmentation strategy that not only reduces the gap with the optimal segmentation, but is also applicable to streaming audio. Toward this goal, since so far no work analyzed in depth the strengths and weaknesses of different audio segmentation methods in the context of direct ST, we first study the behavior of the existing techniques (see §4.2.1). Based on the resulting observations, we propose two variants of an improved hybrid technique. Through experiments in two domains (TED and European Parliament talks) and two target languages (German and Italian), we show that our solutions outperform the others in all conditions, reducing the gap with optimal manual segmentation by at

VAD System	**MuST-C**		**Europarl-ST**		**MuST-C**		**Europarl-ST**	
	BLEU (↑)	**TER (↓)**	**BLEU (↑)**	**TER (↓)**	**BLEU (↑)**	**TER (↓)**	**BLEU (↑)**	**TER (↓)**
	English-German				**English-Italian**			
LIUM	19.55	76.21	15.39	94.06	21.29	67.50	18.88	73.73
WebRTC 3, 30ms	**21.90**	66.96	16.23	89.35	**22.46**	**64.99**	19.85	72.28
WebRTC 3, 20ms	19.48	72.25	14.07	99.32	20.09	68.62	17.35	78.18
WebRTC 2, 20ms	21.87	**66.72**	**18.51**	**78.12**	22.34	66.12	**20.90**	**69.54**

Table 4.4: Results of the VAD systems on MuST-C and Europarl-ST for en-de and en-it.

least 30% compared to VAD systems.

4.4.1 Existing Methods

Before proposing a new segmentation strategy we analyze the quality of existing methods described in §4.2.1 to gain useful insights. The models and training settings used to build them are reported in §4.4.3.

VAD systems. We consider here the same VAD tools and configurations of the previous section (§4.3.3). To better understand the impact of different VADs on translation quality, the tools are compared on MuST-C and Europarl-ST data. Table 4.4 reports preliminary translation results for en-de and en-it. LIUM and the most aggressive WebRTC configuration *(3, 20ms)* are significantly worse than the other two WebRTC configurations. As *(2, 20ms)* achieves comparable BLEU performance to *(3, 30ms)* on MuST-C and better on Europarl-ST, it is used in the rest of the section.

Fixed-length. Fig. 4.3 shows that, with fixed segmentation, translation quality improves with the duration of the segments (slightly for values >=16s) up to 20s, after which it decreases. 20 seconds is the maximum segment length in our training data due to memory limits: we can conclude that longer segments produce better translations, but models can effectively translate only sequences whose length does not exceed the maximum observed in the training set.

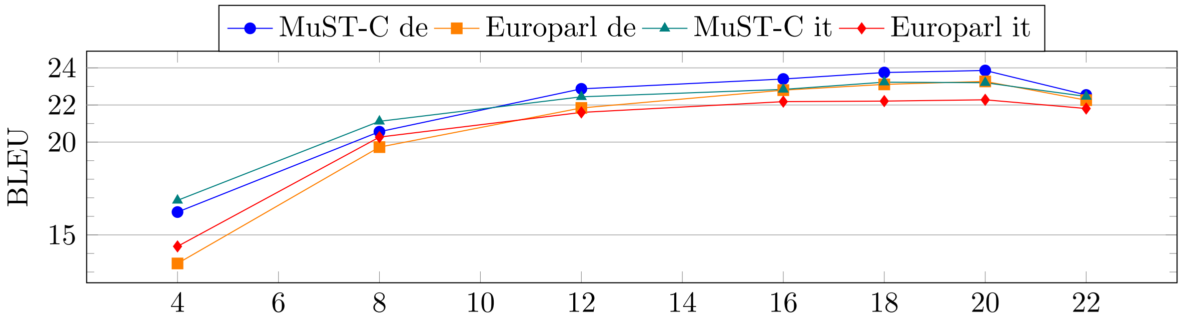


Figure 4.3: BLEU scores with different fixed-length segmentations (in seconds).

SRPOL-like segmentation. For the hybrid method described in §4.2.1, based on the previous considerations drawn from Fig.4.3, in our experiments we set the maximum length threshold to 20s, so that the model is fed with sequences that are not longer than the maximum seen at training time. The resulting segments have an average length of 7-8s.

4.4.2 Proposed hybrid segmentation

Similar to the hybrid method described in §4.2.1, our solution considers both the audio content and the length of the target segments. However, we give more importance to the length of the target segments than to the detected pauses (we motivate this choice in §4.4.4). Specifically, we split on the longest pause in the interval (minimum and maximum length), if any, otherwise we split at maximum length. Maximum and minimum segment lengths are controlled by two hyperparameters (*MAX_LEN* and *MIN_LEN*). Unlike the previously described methods that are based on a *divide and conquer* approach, ours can operate on audio streams, as it does not require the full audio to start the segmentation procedure. Moreover, the latency is controlled by *MAX_LEN* and *MIN_LEN*, which can be tuned to trade translation quality for lower latency.

We tested different values for *MIN_LEN* and we chose 17s for our experiments, because it resulted in the best score on the MuST-C dev set.

Segm. method	**MuST-C en-de**		**Europarl en-de**		**MuST-C en-it**		**Europarl en-it**	
	BLEU (↑)	**TER (↓)**	**BLEU (↑)**	**TER (↓)**	**BLEU (↑)**	**TER (↓)**	**BLEU (↑)**	**TER (↓)**
Manual segm.	27.55	58.84	26.61	60.99	27.70	58.72	28.79	59.16
Best VAD	21.87	66.72	18.51	78.12	22.34	66.12	20.90	69.54
Best Fixed (20s)	23.86	**61.29**	23.27	64.01	23.20	64.24	22.28	64.57
SRPOL-like	22.26	71.10	20.49	77.61	23.12	66.27	23.26	66.19
Pause in 17-20s	**24.39**	61.35	**23.78**	**63.15**	**23.50**	**63.76**	22.86	63.44
+ force split	23.17	66.20	22.52	68.56	23.45	63.79	**24.15**	**63.31**

Table 4.5: Comparison between manual and automatic segmentations: VAD, fixed-length and hybrid approaches.

As in the other methods, and for the same reasons, *MAX_LEN* is set to 20s. The resulting segments have an average length slightly higher than 17s.

We also introduce a variant of this method that enforces splitting on pauses longer than *550ms*. In (Karakanta et al., 2020), this threshold is shown to often represent a *terminal juncture*: a break between two utterances, usually corresponding to clauses. Splitting on such pauses should hence enforce separating different clauses. As a result, segments can be shorter than *MIN_LEN*, but we still ensure they are not longer than *MAX_LEN*. With this variant, the segments are much shorter, as their average length is 8s, similar to that obtained with the SRPOL-like segmentation.

4.4.3 Experimental Settings

We experimented with translation from English speech into two target languages: German and Italian. We compute ST results in terms of BLEU[7] and TER on the test sets of MuST-C and Europarl-ST. In MuST-C the two test sets contain the same audio, while in Europarl-ST there are different recordings.

In our experiments, we use the big system trained on large corpora

[7] Computed with the `multi-bleu.pl` script.

introduced in §3.3. For a complete description of the architecture of the models and the training details, the reader can refer to §3.3.2.

4.4.4 Results

As shown in Table 4.5, fixed-length segmentation always outperforms the best VAD, both in terms of BLEU and TER. This may be surprising, but it confirms previous findings in ASR (Sinclair et al., 2014):[8] also in ST, VAD is more costly and less effective than a naive fixed-length segmentation. Besides, it suggests that the resulting segment length is more important than the precision of the split times. This observation motivates the definition of our proposed techniques. Compared to fixed-length segmentation, the SRPOL-like method provides better results for en-it, but worse for en-de, indicating that the syntactic properties of the source and target languages are a critical factor for audio segmentation (see §4.4.5).

Our proposed method (*Pause in 17-20s* in Table 4.5) outperforms the others on all test sets but Europarl en-it, in which SRPOL-like has a higher BLEU (but worse TER). The version with forced splits on 550ms pauses is inferior to the version without forced splits on the German test sets, but it is on par for MuST-C en-it and superior on Europarl en-it, on which it is the best segmentation overall by a large margin. Moreover, its scores are always better than the ones obtained by the SRPOL-like approach, although the length of the produced segments is similar. These results suggest that, although the best version depends on the syntax and the word order of the source and target languages, our method can always outperform the others in terms of both BLEU and TER. Noticeably, it does not introduce latency, since it does not require the full audio to be available for splitting it, as the SRPOL-like technique does. In particular, averaged on the two domains, our best results (respectively with and without *forced splits*) reduce the

[8] This was also later confirmed by Fukuda et al. (2022).

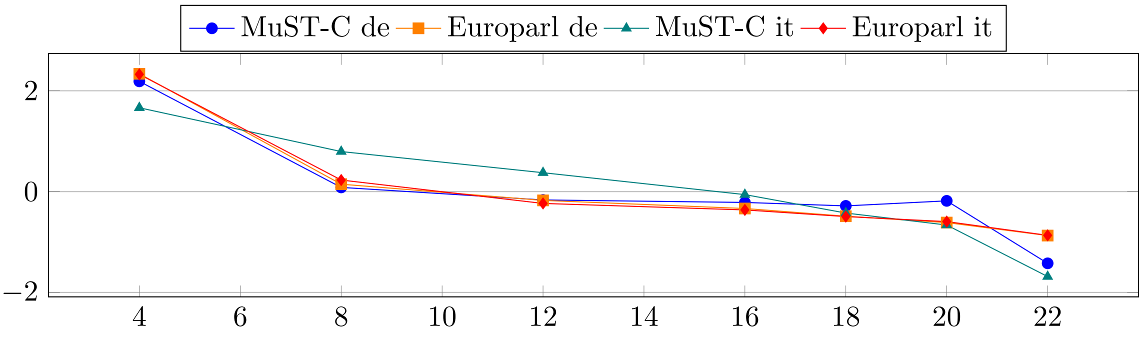


Figure 4.4: Z-score normalized output lengths (number of words) according to the input segments length.

gap with the manual segmentation by 54.71% (en-de) and 30.95% (en-it) compared to VAD-based segmentation.

4.4.5 Analysis

With the goal of understanding the reasons for the different scores, we start our analysis of the outputs produced by the different segmentation methods by inspecting the overall length of the produced translations. In particular, we examine the case of fixed-length segmentation (see Fig. 4.4): in presence of short input segments, the output is longer, while it gets shorter in the case of segments longer than 20s. To understand this behavior, we performed a manual inspection of the German translations produced by fixed-length segmentation with 4s, 20s, and 22s.

The analysis revealed two main types of errors: overly long (*hallucinations*) and overly short outputs. The first type of error occurs when the system is fed with small, sub-sentential segments. In this case, trying to generate well-formed sentences, the system "completes" the translation with text that has no correspondence with the input utterance. The second type of error occurs when the system is fed with segments that exceed the maximum length observed in the training data. In this case, part of the input (even complete clauses, typically towards the end of the utterance) is

(a) Hallucinations with non-speech audio	
Audio	*Music and applause.*
4s segments	**[Chinesisch] [Hawaiianischer Gesang] // Chris Anderson: Du bist ein Idiot. // Nicole: Nein.** *[Chinese] [Hawaiian song] // Chris Anderson: You are an idiot. // Nicole: No.*
(b) Hallucinations with sub-sentential utterances	
Audio	Now, chimpanzees are well-known for their aggression. // (Laughter) // But unfortunately, we have made too much of an emphasis of this aspect (...)
Reference	Schimpansen sind bekannt für ihre Aggressivität. // (Lachen) // Aber unglücklicherweise haben wir diesen Aspekt überbetont (...)
4s segments	**Publikum: Nein.** Schimpansen sind bekannt. // **Ich bin für** ihre Aggression gegangen. // Aber leider haben **wir zu viel Coca-Cola gemacht**. // **Das ist eine** wichtige Betonung dieses Aspekts (...) ***Audience: No.*** *Chimpanzees are known. //* ***I went for*** *their aggression. // But unfortunately* ***we made too much Coca-Cola****. //* ***This is an*** *important emphasis of this aspect (...)*
20s segments	Schimpansen sind bekannt für ihre Entwicklung. // Aber leider haben wir zu viel Schwerpunkt auf diesem Aspekt (...) *Chimpanzees are known for their development. // But unfortunately, we have expressed too much emphasis on this aspect (...)*
(c) Hallucinations and bad translation with sub-sentential utterances	
Audio	(...) where the volunteers supplement a highly skilled career staff, you have to get to the fire scene pretty early to get in on any action.
Reference	(...) in der Freiwillige eine hochqualifizierte Berufsfeuerwehr unterstützten, muss man ziemlich früh an der Brandstelle sein, um mitmischen zu können.
4s segments	(...) **wo die Bombenangriffe auf dem Markt waren. // Man muss bis zu 1.000 Angestellte in die USA, nach Nordeuropa kommen**. *(...)* ***where the bombings were on the market. // You have to come up to 1,000 employees in the USA, to Northern Europe****.*
20s segments	(...) in der die Freiwilligen ein hochqualifiziertes Karriere-Team ergänzen, muss man ziemlich früh an die Feuerszene kommen, um in irgendeiner Aktion zu gelangen. *(...) where the volunteers complement a highly qualified career team, you have to get to the fire scene pretty early in order to get into any action.*
(d) Final portions of long segment ignored	
Audio	But still it was a real footrace against the other volunteers to get to the captain in charge to find out what our assignments would be. // When I found the captain, (...)
Reference	Aber es war immer noch ein Wettrennen gegen die anderen Freiwilligen **um den verantwortlichen Hauptmann zu erreichen und herauszufinden was unsere Aufgaben sein würden**. // Als ich den Hauptmann fand (...)
22s segments	(...) Es war immer noch ein echtes Fussrennen gegen die anderen Freiwilligen. // Als ich den Kapitän fand, (...) *(...) It was still a real footrace against the other volunteers. // When I found the captain, (...)*

Table 4.6: Translations affected by errors caused by too short – *(a)*, *(b)*, *(c)* – or too long – *(d)* – segments. The symbol "//" refers to a break between two segments. The breaks might be located in different positions in the different segmentations. Over-generated – in examples *(a)*, *(b)*, *(c)* – and missing – in *(d)* – content is marked in **bold** respectively in the system outputs and in the reference.

not realized in the final translation.

Table 4.6 provides examples of all these phenomena. The first three examples showcase hallucinations in short (4s) segments, while the last one shows an incomplete translation of a long (22s) segment. In particular:

(a) shows the generation of text not related to the source when the audio contains only noise or silence (e.g. at the beginning of a TED talk recording).

(b) presents the addition of non-existing content in the translation of a sub-sentential segment.

(c) is related to a sub-sentential utterance as well, but in this case the output of the system is affected by both hallucinations and poor translation quality due to the lack of enough context.

(d) reports a segment whose last portion is ignored.

The length of the generated outputs also helps to understand the different results obtained by the variants of our method on the two target languages. Indeed, the introduction of forced splits (*+ force split*) produces audio segments that are much shorter (∼8s vs ∼17s) and hence, according to the previous consideration, the resulting translation is overall longer. For German, the difference in terms of output length is high ($>$ 8.5%), while for Italian it is much lower (4.33% on MuST-C and 2.49% on Europarl-ST). So, the German results are penalized by the additional hallucinations, while, for Italian translations, the beneficial separation of clauses delimited by terminal juncture dominates.

This different behavior relates to the different syntax of the source and target languages. Indeed, translating from English (an SVO language) into German (an SOV language) requires long-range re-orderings (Gojun and Fraser, 2012; Navrátil et al., 2012), which can also span over sub-clauses. The Italian phrase structure, instead, is more similar to English. This is confirmed by the shifts counted in TER computation, which are 20% more in German than in Italian. Moreover, in Italian their number does not change between our method with and without forced splits, while in German the version with forced splits has 5-10% more shifts.

4.4.6 Summary

Our study on reducing the quality drop caused by suboptimal segmentation of the audio at inference time has progressed in this section with a comparison of different segmentation techniques for direct ST. Despite its wide adoption, VAD-based segmentation resulted to be underperforming. We showed that the length of audio segments is a crucial factor to obtain good translations and that the best segmentation approach depends on the structural similarity between the source and target languages. In particular, we demonstrated that the resulting segments should be neither longer than the maximum length of the training samples nor too short (especially when the target language has a different structure). Inspired by these findings, we proposed two variants of a hybrid method that significantly improve on different test sets and languages over the VAD baseline and other techniques, reducing by at least 30% the gap with optimal manual segmentation. In addition, our approach was designed to be also applicable to audio streams and to allow controlling latency, hence being suitable even for online use cases. All in all, our methods improve the translation quality while keeping low the computational cost and controlling the latency introduced, in line with the spirit of this thesis. The next section combines our segmentation strategy with the techniques proposed in §4.3 to assess their complementarity and overall effectiveness.

4.5 Combination of Approaches

We now combine the two approaches presented in §4.3 and §4.4 to investigate whether their benefits are cumulative or not. Moreover, as Tsiamas et al. (2022b) compare their newly proposed SHAS method (see §4.2.1) with other segmentation methods only using models trained on well-formed sentence-utterance pairs, we complement their experiments by validating

Model	**BLEU (↑)**	**TER (↓)**
BASE	24.39	61.35
FINE-TUNE	**25.39**	57.64
TGT PAR + REG	24.92	**57.41**
VAD (TGT PAR + REG)	23.91	58.95

Table 4.7: Comparison between fine-tuned and context-aware models with hybrid audio segmentation on the en-de section of MuST-C.

their findings also on models robust to automatic segmentation. In particular, we investigate whether *i)* model robustness brings benefits also with audio segmented with SHAS, and *ii)* the gap between SHAS and other segmentation methods is closed or not by the adoption of robust models.

Our experiments show that the two approaches (model robustness and audio segmentation) account for complementary gains, both contributing to reduce the detrimental effect of the audio segmentation mismatch between training and inference data. However, this finding does not hold when SHAS is used. In this case, the model robustness to automatic segmentation results irrelevant due to the high quality of the automatic segmentation.

4.5.1 Model Robustness with Hybrid Audio Segmentation

We evaluate the models proposed in §4.3 on audio segmented with the hybrid method introduced in §4.4 to assess whether their benefits are complementary. Table 4.7 presents the results.

First, both approaches to increase model robustness to automatic segmentation (fine-tuning and context-based) significantly improve the scores (+0.5/1.0 BLEU and -3.7/3.9 TER). Moreover, there are large gains (+1.0/1.5 BLEU and -1.3/1.5 TER) compared to the best scores obtained using VAD tools to segment the audio. We can conclude that the two approaches are complementary in reducing the gap with optimal audio segmentation, limited to only 2.16 BLEU in the best scenario.

Model	**Hybrid**		**SHAS**	
	tst-COMMON	iwslt2020	tst-COMMON	iwslt2020
Conformer	27.4	23.8	30.3	26.4
Conformer + resegm. fn	29.1	25.0	29.9	26.2

Table 4.8: BLEU scores of Hybrid and SHAS audio segmentation methods of the models with and without fine-tuning on re-segmented data (*resegm. fn*) on the MuST-C v2 tst-COMMON and the IWSLT2020 test set.

Comparing the two techniques to increase model robustness, the fine-tuning and the context-aware models show very similar results in this scenario, with differences that are not statistically significant and are not coherent across metrics and test sets. Although this finding may seem to contradict previous results, the behavior is explained by the previously-mentioned observation that in presence of long segments (as those produced by our hybrid method) the beneficial contribution of the previous segment is low. In light of these results and the fact that the fine-tuning is a simpler approach, this method is considered in the experiments of the next section.

4.5.2 Comparison with SHAS

This section compares our hybrid method with the recent SHAS segmentation in the condition in which the ST models are adapted to be robust to a suboptimal audio segmentation. For this comparison, we leverage the state-of-the-art Conformer models used in §3.6. For the training details, the reader can refer to §3.6.3.

Table 4.8 compares our hybrid approach with the recent SHAS method, with and without fine-tuning. First, we notice that the SHAS segmentation improves over the Hybrid one, with gains from 0.8 to 2.9 BLEU. Secondly, we see that the fine-tuning on re-segmented data – useful with the Hybrid segmentation – becomes useless if using SHAS. In fact, the best overall results are obtained using SHAS on a model that is not fine-tuned on resegmented data, which scores 30.3 BLEU on the MuST-C v2 tst-COMMON and 26.4 BLEU on the IWSLT 2020 test set. As such, we can conclude that fine-tuning on resegmented data is not needed if the audio is segmented with SHAS.

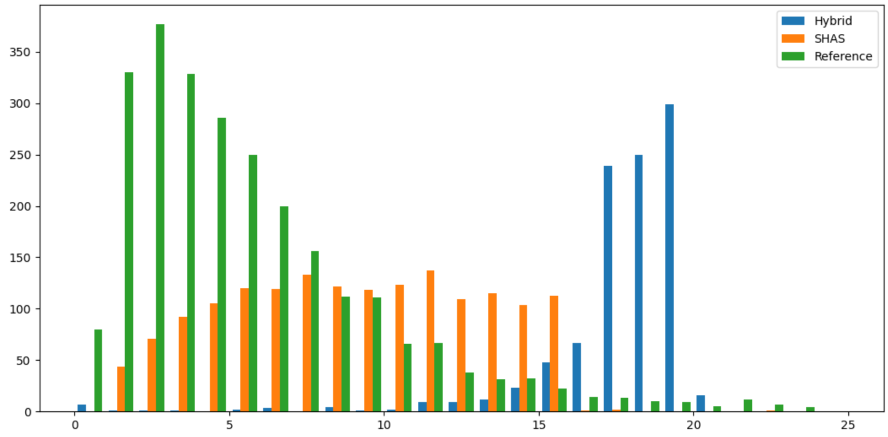


Figure 4.5: Histogram of sequence lengths (in seconds) of the segments generated by our Hybrid method and SHAS, compared with the reference, on the tst-COMMON of MuST-C en-de v2.

To delve deeper into the comparison between our audio segmentation method and SHAS, we analyze the lengths of the utterances obtained with the two audio segmentation methods. Figure 4.5 compares the two methods with the reference segmentation released for the MuST-C v2 test set. We can notice that the distribution of segment lengths is very different between all methods: our Hybrid method has a peaky distribution between 17 and 20 seconds, as expected, with a long tail toward 0 seconds, corresponding to the last parts of the TED talks; SHAS has a quite flat distribution between 1 and 16 seconds (the default max segment length configured in SHAS), with few outliers that go up to 23s; lastly, the segmented in the reference display a higher variability, although most of them are shorter than 10 seconds, with a long tail of longer segments. These differences are also reflected in the mean and variance of the methods. Our Hybrid solution has the highest mean (17.7s) and lowest variance (6.6), while the reference has the lowest mean (5.8s) and highest variance (22.2). SHAS positions in the middle, although it is closer to the reference, with an average segment length of 9.1 seconds and 16.0 variance.

4.5.3 Summary

This section has shown the effect of combining our approaches for mitigating the impact of the audio segmentation mismatch between training and inference data. We have seen that the benefits brought by a model robust to a different segmentation vary according to the quality of the audio segmentation. With our hybrid audio segmentation method, model robustness contributes to closing the gap with optimal segmentation. The comparison of our segmentation technique with the recent SHAS revealed that SHAS leads to better results and eliminates the need for dedicated adaptations of the ST model. However, we reiterate that SHAS has limitations (see §4.2.1) that prevent its adoption in a streaming scenario. Thus, our hybrid method, complemented by the proposed fine-tuning on resegmented data, still represents a valid alternative in specific scenarios.

4.6 Conclusions

In this chapter, we confronted the topic of automatic segmentation of long speeches, which is a preparatory step for translating with direct ST models. After the assessment of the huge quality drop due to the distributional shift between the manually segmented training data and VAD-segmented inference data, we addressed the problem with two approaches. On one side, our first contribution is the introduction of two methods to create models robust to the automatic segmentation of the audio. These methods are *i)* a fine-tuning on randomly-segmented training data, and *ii)* a context-aware architecture that attends to the previous segment as contextual information. On the other side, we pursued the goal of limiting the distributional shift between training and inference data by means of a novel hybrid solution for the automatic segmentation of the audio. Lastly, we combined the two approaches, showing that they bring complementary advantages, restraining the gap with the optimal segmentation to only 2.16 BLEU, from the 5.65 BLEU of the best VAD tool. This chapter concludes our activities on improving the quality of direct ST systems, even in the realistic scenario of unsegmented audio. The results achieved lay the foundations for the in-depth analysis of the next two chapters on specific capabilities of direct ST models, which are crucial for their adoption in production. Along this line, Chapter 5 focuses on gender bias, while Chapter 6 explores the translation and recognition of salient elements, in particular named entities, in the output of direct ST systems.

Chapter 5

Gender Bias

5.1 Introduction

The previous chapters described the efforts toward high-quality direct ST systems to make them ready for real applications. However, overall translation quality is not the only factor that should be considered to determine whether the technology is ready or not for production. Indeed, the widespread use of language technologies has motivated growing interest on their social impact (Hovy and Spruit, 2016; Blodgett et al., 2020), with gender bias representing a major cause of concern (Costa-jussà, 2019; Sun et al., 2019). As regards translation tools, focused evaluations have exposed that even state-of-the-art ST – and MT – models do in fact overproduce masculine references in their outputs (Cho et al., 2019; Bentivogli et al., 2020), except for feminine associations perpetuating traditional gender roles and stereotypes (Prates et al., 2020; Stanovsky et al., 2019). The problem is particularly critical when translating from genderless languages[1] (e.g., Finnish, Turkish) or notional gender languages[2] (e.g., Danish, English) into

[1] Languages in which the gender-specific repertoire is at its minimum, only expressed for basic lexical pairs, usually kinship or address terms (e.g., in Finnish *sisko*/sister vs. *veli*/brother).

[2] Languages that display a system of pronominal gender (*she/he*, *her/him*). English also hosts some marked derivative nouns (*actor*/*actress*) and compounds (*chairman*/*chairwoman*).

grammatical gender languages[3] (e.g., Arabic, Spanish). In this case, the richer morphology of the target language requires automatic models to generate outputs that overtly assign a gender even without any explicit indication in the source.

Most works identified *data* as the primary source of gender asymmetries. Accordingly, many pointed out the misrepresentation of gender groups in datasets (Vanmassenhove et al., 2018; Garnerin et al., 2019), focusing on the development of data-centered mitigating techniques (Zmigrod et al., 2019; Saunders and Byrne, 2020). However, data are not the only factor contributing to gender bias (Shah et al., 2020). Neural networks rely on easy-to-learn shortcuts or "cheap tricks" (Levesque, 2014), as picking up on spurious correlations offered by training data can be easier for machines than learning to actually solve a specific task. What is "easy to learn" for a model depends on the *inductive bias* (Sinz et al., 2019; Geirhos et al., 2020) resulting from architectural choices, training data, and learning rules. As such, technical components can exacerbate the problem (Vanmassenhove et al., 2019) and architectural changes can contribute to its mitigation (Costa-jussà et al., 2020). Also, "taken-for-granted" approaches that come with high overall translation quality may actually be detrimental when it comes to gender bias (Roberts et al., 2020). In addition, unlike cascade architectures, direct ST models may leverage the acoustic properties of the audio input (e.g., speaker's fundamental frequency) to determine the gender of the speaker, as they are almost always aligned in training corpora. However, relying on perceptual markers of speakers' gender is not the best solution for all users (e.g., transgenders, children, vocally-impaired people).

For these reasons, we believe that the technological development of

[3] Languages in which each noun pertains to a class such as masculine, feminine, and neuter (if present). Although for most inanimate objects gender assignment is only formal, for human referents gender markings are assigned on a semantic basis. Several parts of speech besides the noun (e.g., verbs, determiners, adjectives) carry gender inflections, according to a system of morphosyntactic agreement.

new solutions should also account for the biasing effects they can have. With this spirit, this chapter investigates the gender bias introduced by the techniques leveraged to train direct ST models, eventually proposing mitigating strategies. To this aim, we first study different solutions to exploit external metadata indicating the gender of the speaker[4] to control the gender realizations in the translation (§5.3). Then, we inspect the effect of different word segmentation strategies on gender translation (§5.4). Along the same line, we carry out a multifaceted evaluation of systems with different segmentation strategies and trained on different amounts of data, assessing their capabilities on different parts of speech (POS) and their coherence in morphosyntactic agreement chains (§5.5). We conclude by complementing our study on KD from MT systems by analyzing the effects it can have on gender translation, especially on first-person references (§5.6).

The contributions of this chapter include: *i)* comparing different solutions that integrate the external knowledge about the speaker's gender and control the generated translation accordingly; *ii)* showing that the widespread BPE tokenization of the text exacerbates gender disparities in the training data and proposing a solution to maintain the same translation quality while limiting gender bias; *iii)* introducing a fine-grained evaluation of gender bias in ST systems, thanks to which we individuate in *nouns* the most biased POS and assess the almost perfect ability of ST systems in respecting morphosyntactic agreement rules; *iv)* unveiling the negative effects of KD from MT on gender bias and how to mitigate them.

[4] This information may be available in many situations. For instance, before a talk the speaker can be asked for their gender.

5.2 Related Works

After an overview of the (few) works on gender bias in ST (§5.2.1), we present studies aimed at controlling gender translation in MT (§5.2.2), as our solutions (described in §5.3) are the first to do so in ST. We then provide and overview of the different word segmentation strategies (§5.2.3), whose effect on gender bias is analyzed in §5.4.

5.2.1 Gender Bias in ST

Despite the importance of the topic, few works analyzed gender bias in direct ST. Bentivogli et al. (2020) introduced MuST-SHE, a gender-sensitive benchmark available for English→{French, Italian, Spanish}. Built on naturally occurring instances of gender phenomena retrieved from MuST-C, it allows evaluating gender translation on qualitatively differentiated and balanced masculine/feminine forms. MuST-SHE contains two main categories of sentences: category 1 refers to words whose correct translation depends on the gender of the speaker (as in *I've never* ***been*** *there*, which is translated in Italian as *Non sono mai* ***stata/stato*** *lì* according to whether it is uttered by a woman or a man), and category 2 to those in which the gender of the translated words is derived by a pronoun or another referent present in the sentence (as in *He/She work as a* ***doctor***, which is translated in Italian as *Lui/Lei lavora come* ***dottore/dottoressa***). Thanks to this resource, Bentivogli et al. (2020) compared gender translation performance of cascade and direct ST, proving that the latter has an advantage when it comes to speaker-dependent gender translation (category 1), since it can leverage acoustic properties from the audio that have a strong correlation with correct gender realizations in the test set.

Costa-jussà et al. (2022) extend the popular challenge test set WinoMT by Stanovsky et al. (2019) to ST, recording 3,888 English sentences uttered by an American female speaker. This resource can be useful to diagnose gender stereotyping at scale when translating into grammatical gender languages, as it consists of synthetic sentences with the same structure and a pre-selected occupational lexicon, in which a pronominal coreference determines the gender of the referent. The gender information of the evaluated terms is always explicit in the content of the utterances and the gender of the speaker does not play any role: in this sense, the test set is similar to category 2 of MuST-SHE. Through experiments on 4 language pairs, Costa-jussà et al. (2022) demonstrate that ST systems exhibit a disproportionate production of masculine references in their outputs, except for feminine associations with traditional gender roles and characteristics.

Lastly, Zanon Boito et al. (2022) annotate the French talks of the mTEDx corpus (Salesky et al., 2021) and evaluate the overall performance of models obtained by fine-tuning pre-trained models such as wav2vec 2.0 (Baevski et al., 2020) on the ASR and ST downstream tasks. They both pre-train and fine-tune wav2vec 2.0 with varying amounts of data of speakers of each gender and find out that gender-specific pre-training causes performance degradation, while balanced pre-training does not imply gender fairness. They do not focus, though, on the gender realizations in the output.

All these works assess the presence of gender bias in direct ST systems without focusing on the impact of specific technical choices or components, i.e. on the algorithmic bias, as we do throughout this chapter. In addition, to the best of our knowledge, we are the first to propose mitigation strategies in direct ST.

5.2.2 Controlling Speaker Gender in Translation

While no previous work in ST was dedicated to controlling the gender realization of words referred to the speaker, in the related field of MT a few approaches have been employed to make neural MT systems aware of speakers' gender. We divide them into two categories: black-box methods, which intervene only in the inference phase, and gender tagging methods, which mitigate gender bias through architectural changes and dedicated training procedures.

Black-box methods. Moryossef et al. (2019) attempt to control the production of feminine references to the speaker and numeral inflections (plural or singular) for the listener(s) in an English-Hebrew spoken language setting. To this aim, they rely on a short construction, such as "*she* said to *them*", which is prepended to the source sentence and then removed from the MT output. Their approach is simple, it can handle two types of information (gender and number) for multiple entities (speaker and listener), and improves the ability of systems in generating feminine target forms. This solution is hardly applicable to ST, though, as prepending information to the source is not trivial due to the audio modality. Habash et al. (2019) and Alhafni et al. (2020) confront the problem of speaker's gender agreement in Arabic with a post-processing component that re-inflects 1st person references into masculine/feminine forms. In (Alhafni et al., 2020), the preferred gender of the speaker and the translated Arabic sentence are fed to the component, which re-inflects the sentence in the desired form. In (Habash et al., 2019) the component can be: *i)* a two-step system that first identifies the gender of 1st person references in an MT output, and then re-inflects them in the opposite form; *ii)* a single-step system that always produces both forms given an MT output. However, the implementation of the re-inflection component was made possible by the Arabic Parallel Gender Corpus (Habash et al., 2019), which demanded an expensive work of manual data creation, hence being hardly extendible to other language pairs.

Gender tagging. To improve the generation of speaker's referential markings, Vanmassenhove et al. (2018) prepend a gender tag (M or F) to each source sentence, both at training and inference time. This approach was inspired by one-to-many multilingual NMT systems (Johnson et al., 2017), in which a single model translates from a source into many target languages by means of a *target-forcing* mechanism. The solution proves useful to handle morphological agreement when translating from English into French. Similar to our work in §5.3, this approach requires additional metadata regarding the speakers' gender. Elaraby et al. (2018) avoid this need by defining a comprehensive set of cross-lingual gender agreement rules based on POS tagging. In this way, they identify speakers' and listeners' gender references in an English-Arabic parallel corpus, which is consequently labeled and used for training. The idea can be adapted for other languages and scenarios by creating new dedicated rules. However, in realistic deployment conditions where reference translations are not available, gender information still has to be externally supplied as metadata at inference time. Stafanovičs et al. (2020) and Saunders et al. (2020) explore the use of word-level gender tags. While Stafanovičs et al. (2020) just report a gender translation improvement, Saunders et al. (2020) rely on the expanded version of WinoMT to identify a problem concerning gender tagging: it introduces noise if applied to sentences with references to multiple participants, as it pushes their translation toward the same gender. Saunders et al. (2020) also include a first non-binary exploration of neutral translation by exploiting an artificial dataset, where neutral tags are added and gendered inflections are replaced by placeholders. The results are however inconclusive, most likely due to the small size and synthetic nature of their dataset. Both solutions are not applicable to ST, though, due to the different source modality, as they require identifying speaker-referred words in the source.

5.2.3 Word Segmentation

Although early attempts in neural MT employed word-level sequences (Sutskever et al., 2014; Bahdanau et al., 2015), the need for open-vocabulary systems able to translate rare/unseen words led to the definition of several word segmentation techniques. Currently, the statistically motivated approach based on BPE (Sennrich et al., 2016; Kudo and Richardson, 2018) represents the *de facto* standard in MT. Recently, its superiority to character-level segmentation (Costa-jussà and Fonollosa, 2016; Chung et al., 2016; Lee et al., 2017) has been also proved in the context of ST (Di Gangi et al., 2020a). However, depending on the languages involved in the translation task, the data conditions, and the linguistic properties taken into account, BPE greedy procedures can be suboptimal. By breaking the surface of words into plausible semantic units, linguistically motivated segmentations (Smit et al., 2014; Ataman et al., 2017) were proven more effective for low-resource and morphologically-rich languages (e.g., agglutinative languages like Turkish), which often have a high level of sparsity in the lexical distribution due to their numerous derivational and inflectional variants. Moreover, fine-grained analyses comparing the grammaticality of character, morpheme and BPE-based models exhibited different capabilities. Sennrich (2017) and Ataman et al. (2019) showed the syntactic advantage of BPE in managing several agreement phenomena in German, a language that requires resolving long-range dependencies. In contrast, Belinkov et al. (2020) demonstrated that while subword units better capture semantic information, character-level representations perform best at generalizing morphology, thus being more robust in handling unknown and low-frequency words. Indeed, using different atomic units does affect the model ability to handle specific linguistic phenomena. All these works analyze the different segmentation methods in terms of their impact on overall quality. However, whether low gender translation accuracy can also be to a certain extent considered a by-product of certain text-segmentation algorithms has not been investigated. Therefore, we aim at filling this gap in §5.4.

5.3 Gender-aware Systems

As seen in §5.2.1, by translating speech audio data without intermediate transcription, direct ST models are able to infer speakers' gender from their vocal characteristics, which are otherwise lost in the cascade framework. Although such ability proved to be useful for gender translation, since female speakers (and associated feminine marked words) are less frequent within the training corpora, direct ST nonetheless tends towards a masculine default just like its cascade counterpart, as well as MT, and numerous other natural language processing applications. Moreover, direct ST systems that exclusively rely on vocal biometric features as a gender cue can be unsuitable and potentially harmful for certain users (e.g., transgenders, children, vocally-impaired people).

Going beyond speech signals, in this section we compare different approaches to inform direct ST models about the speaker's gender and test their ability to control gender translation from English into Italian and French. Toward this objective, we annotated MuST-C with speakers' gender information and explored different techniques to exploit such information in direct ST to go beyond a potentially harmful exploitation of speakers' vocal traits. The proposed techniques are compared, in terms of both overall translation quality and accuracy in the translation of gender-marked words, against a "pure" model that solely relies on the speakers' vocal characteristics for gender disambiguation.

In light of the above, our contributions are *i)* the manual annotation of the TED talks contained in MuST-C with speakers' gender information, based on the personal pronouns found in their TED profile[5], and *ii)* the first comprehensive exploration of different approaches to mitigate gender bias in direct ST, depending on the potential users, the available resources, and the architectural implications of each choice.

Experiments carried out on English-Italian and English-French show that, on both language directions, our gender-aware systems significantly outperform "pure" ST models in the translation of gender-marked words (up to 30 points in accuracy) while preserving overall translation quality. Moreover, our best systems learn to produce feminine/masculine gender forms regardless of the perceptual features received from the audio signal, offering a solution for cases where relying on speakers' vocal characteristics is detrimental to a proper gender translation.

5.3.1 Speakers' Gender Annotation

The assumption that the gendered forms expected in translation align with speakers' vocal characteristics is detrimental to different groups of users. As such, a better alternative is to directly enforce and control the correct gender of the speaker if it is known in advance. To enable researchers in ST to study solutions for this scenario, we decided to build a training ST corpus explicitly annotated with gender information. To this aim, rather than building a new resource from scratch, we opted for adding an annotation layer to MuST-C, which has been chosen over other existing corpora for the following reasons: *i)* it is currently the largest freely available multilingual corpus for ST, *ii)* being based on TED talks it is the most compatible one with MuST-SHE, *iii)* TED speakers' personal information is publicly available and retrievable on the TED official website.[6]

[5] The resource, MuST-Speakers, is released under a CC BY NC ND 4.0 International license, and is freely downloadable at `https://ict.fbk.eu/must-speakers/`.

[6] Available at `https://www.ted.com/speakers/`

Following the MuST-C talk IDs, we have been able to *i)* automatically retrieve the speakers' name, *ii*) find their associated TED official page, and *iii*) manually label the personal pronouns used in their descriptions. Though time-consuming, such manual retrieval of information is preferable to automatic speaker gender identification for the following reasons. First, since automatic methods based on fundamental frequency are not equally accurate across demographic groups (e.g., women and children are hard to distinguish as their pitch is typically high – Levitan et al. 2016), manual assignment prevents from incorporating gender misclassifications in our training data. Second, biological essentialist frameworks that categorize gender based on acoustic cues (Zimman, 2020) are especially problematic for transgender individuals, whose gender identity is not aligned with the sex assigned at birth based on designated anatomical/biological criteria (Stryker, 2008).

Differently, following the guidelines in (Larson, 2017), we do not want to run the risk of making assumptions about speakers' gender identity and introducing additional bias within an environment that has been specifically designed to inspect gender bias. By looking at the personal pronouns used by the speakers to describe themselves, our manual assignment instead is meant to account for the gender linguistic forms by which the speakers accept to be referred to in English (GLAAD, 2007), and would want their translations to conform to. We stress that gendered linguistic expressions do not directly map to speakers' self-determined gender identity (Cao and Daumé III, 2020). We therefore make explicit that, when talking about speakers' gender, we refer to their accepted linguistic expression of gender rather than their gender identity.

Focusing on the two language pairs of our interest, 2,294 different speakers described via *he/she* pronouns[7] are represented in both en-it and en-fr.

[7] It is important to point out that some individuals do not neatly fall into the female/male binary

	Talks M	Talks F	Hours M	Hours F	Segments M	Segments F
en-it	1,569	725	316	136	178,841	71,877
en-fr	1,569	725	327	151	189,742	81,527

Table 5.1: Statistics for MuST-C data with gender annotation. The number of segments and hours varies over the two language pairs due to the different pre-processing of MuST-C data.

Their male/female[8] distribution is unbalanced, as shown in Table 5.1, which presents the number of talks, as well as the total number of segments and the corresponding hours of speech.

5.3.2 ST Systems

For our experiments, we build three types of direct systems. One is the *base* system, a state-of-the-art model that does not leverage any external information about the speaker's gender (§5.3.2). The others are two gender-aware systems that exploit speakers' gender information in different ways: *multi-gender* (§5.3.2) and *specialized* (§5.3.2). All the models share the same architecture, a Transformer adapted to ST, analogous to that used in previous experiments (e.g., §3.3, §4.4), with logarithmic distance penalty.

(gender fluid, non-binary) or may even not experience gender at all (a-gender) (Richards et al., 2016; Schilt and Westbrook, 2009; GLAAD, 2007), possibly preferring the use of singular *they* or other neopronouns. Within MuST-C, speakers with *they* pronoun have been encountered, but MuST-C human-reference translations do not exhibit linguistic gender-neutralization strategies, which are difficult to fully implement in languages with grammatical gender (Lessinger, 2020). Note that, because of such inconsistency and the very limited number of cases, these instances were not used for training. Our experiments therefore focus on binary linguistic forms. By design, some sparse talks with multiple speakers of different genders were also excluded. Detailed information about all MuST-C speakers and corresponding talks can be found in the resource release at `ict.fbk.eu/must-speakers`.

[8] Some authors distinguish female/male for sex and woman/man for gender (among others Larson 2017). For the sake of simplicity, in our study we use female/male to respectively indicate those speakers whose personal pronouns are *she/he*.

Base ST Model

We are interested in evaluating and improving gender translation on strong ST models that can be used in real-world contexts. As such, our base, gender-unaware model is trained with the goal of achieving state-of-the-art performance on the ST task. To this aim, we rely on data augmentation and knowledge transfer techniques (see §2.3.2 and §3.3). In particular, we use SpecAugment, time stretch, and synthetic data generation, and we transfer knowledge both from ASR and MT through, respectively, component initialization and KD.

The ST-model encoder is initialized with the encoder of an English ASR model with a lower number of encoder layers (the missing layers are initialized randomly, as well as the decoder). This ASR model is trained on Librispeech, Mozilla Common Voice, How2, TEDLIUM-v3, and the utterance-transcript pairs of the ST corpora – Europarl-ST and MuST-C. These datasets are either gender unbalanced or do not provide speakers' gender information apart from Librispeech, which is balanced in terms of female/male speakers (Garnerin et al., 2020). However, since these speakers are just book narrators, first-person sentences do not really refer to the speakers themselves. For both en-it and en-fr, the MT model is trained on the OPUS datasets.

The ST model is trained in three consecutive steps, following the methodology introduced in §3.3. In the first step, we use the synthetic data obtained by pairing ASR audio samples with the automatic translations of the corresponding transcripts. In the second step, the model is trained on the ST corpora. In these first two steps, we use the KD loss function. Finally, in the third step, the model is fine-tuned on the same ST corpora using label-smoothed cross entropy.

Multi-gender Systems

As anticipated in §5.2.2, the idea of "multi-gender" models, i.e. models informed about the speaker's gender with a tag prepended to the source sentence, was introduced by Vanmassenhove et al. (2018) and Elaraby et al. (2018) in MT. With this mechanism, ST multi-gender systems are fed not only with the input audio, but also with a tag (*token*) representing the speaker's gender. This *token* is converted into a vector through learnable embeddings. This approach has two main potential advantages: *i)* a single model supports both male and female speakers (which makes it particularly appealing for real-world application scenarios), and *ii)* each gender direction can benefit from the data available for the other, potentially learning to produce words that would have never been seen otherwise (*transfer learning*). Regarding the several options to supply the model with the additional gender information, we do not follow the approach of Vanmassenhove et al. (2018) and Elaraby et al. (2018), since it is dedicated to MT. Instead, we consider those that obtained the best results in multilingual direct ST (Di Gangi et al., 2019d; Inaguma et al., 2019), namely:

Decoder prepending. The gender token replaces the *BOS* token[9] (beginning-of-sentence). This is the token added in front of the previously generated tokens fed to the autoregressive decoder.

Decoder merge. The gender embedding is added to all the word embeddings representing the generated tokens in the decoder input.

Encoder merge. The gender embedding is added to the Mel-filter-bank sequence representing the source speech given as input to the encoder.

[9] In our implementation, the *EOS* (end-of-sentence) token is actually used in place of the *BOS*, which does not exist.

In all cases, model weights are initialized with those of the *Base* models. The only randomly-initialized parameters are those of the gender embeddings.

Gender-specialized Systems

In this approach, two different gender-specific models are created. Each model is initialized with the weights of the *Base* model and then fine-tuned only on samples of the corresponding speaker's gender. This solution has the drawback of a higher maintenance burden than the multi-gender one, as it requires the training and management of two separate models. Moreover, no transfer learning is possible: although each model is initialized with the base model trained on all the data and the low learning rate used in the fine-tuning prevents catastrophic forgetting (Mccloskey and Cohen, 1989), data scarcity conditions for a specific gender are likely to lead to lower performance on that direction.

5.3.3 Gender-balanced Validation Set

To train our gender-aware models, we do not rely on the standard MuST-C validation set, as it reflects the same gender-imbalanced distribution found in the training data. We therefore created a new specifically designed validation set composed of 20 talks. Unlike the standard MuST-C validation set, it contains a balanced number of female/male speakers, thus avoiding the reward of potentially biased behaviors. This new resource is released under a CC BY NC ND 4.0 International license, and is freely downloadable at `https://ict.fbk.eu/must-c-gender-dev-set/`.[10]

[10] To ease future research on gender bias in ST for the three language pairs represented in MuST-SHE (en-it, en-fr, en-es), the validation set is also available for en-es.

5.3.4 Experimental Setting

Multi-gender (§5.3.2) and gender-specialized models (§5.3.2) are initialized with the weights of the base model and are then fine-tuned on the MuST-C gender-labeled dataset. Since, as seen in §5.3.1, this dataset shows a quite skewed male/female speaker distribution (approximately 70%/30%), we test both approaches in two different data conditions: *i)* balanced (*-Bal), where we use all the female data available together with a random subset of the male data, and *ii)* unbalanced (*-All) where all the MuST-C data available are exploited. It must be noted that there are differences between the two approaches on the usage of data. In the specialized approach, since we have two separate systems, the one which is fine-tuned with talks by female speakers remains the same in both data conditions. Differently, in the multi-gender approach, which is trained on both genders together, all the training mini-batches contain the same number of samples for each gender. Thus, when all MuST-C data are used, the female gender pairs – which are underrepresented – are over-sampled.

5.3.5 Evaluation Method

For our experiments, we rely on MuST-SHE. By design, each segment in the corpus requires the translation of at least one English gender-neutral word into the corresponding masculine/feminine target word(s) to convey a referent's gender. With the intent to evaluate our gender-aware ST models on speaker-dependent gender phenomena, we focus on category 1 of MuST-SHE (see §5.2.1).

An important feature of MuST-SHE is that, for each reference translation, an almost identical "wrong" reference is created by swapping each annotated gender-marked word into its opposite gender (e.g., *I have been* uttered by a woman is translated into the correct Italian reference *Sono stat**a***, and into the wrong reference *Sono stat**o***). The idea behind gender-swapping is that the difference between the scores computed against the "correct" and the "wrong" reference sets captures the ability of the systems to handle gender translation. However, relying on these scores does not allow distinguishing between those cases where the system "fails" by producing a word different from the one present in the references (e.g., *andat** in place of *stat**) and failures specifically due to the wrong realization of gender (e.g., *stat**o*** in place of *stat**a***).

Thus, we introduce a more informative evaluation. First, we calculate the **term coverage** as the proportion of gender-marked words annotated in MuST-SHE that are actually generated by the system, on which the accuracy of gender realization is therefore *measurable*. Then, we define **gender accuracy** as the proportion of correct gender realizations among the words on which it is *measurable*. Our evaluation method has several advantages. On one side, *term coverage* unveils the precise amount of words on which gender realization is measurable. On the other, *gender accuracy* directly informs about performance on gender translation and related gender bias: scores below 50% indicate that the system produces the wrong gender more often than the correct one, thus signalling a particularly strong bias. Gender accuracy has the further advantage of informing about the margins for improvement of the systems.

5.3.6 Results

Table 5.2 presents overall results in terms of BLEU scores on the MuST-SHE test set. Both language directions show the same trend.

First, the MT systems used by the ST models for KD achieve by far the highest performance. This is expected since the ST task is more complex and MT models are trained on larger amounts of data. However, all our ST results are competitive compared to those published for the two target

	en-it	**en-fr**
	BLEU	BLEU
MT for KD	33.59	39.61
Base	27.51	34.25
Multi-DecPrep-Bal	26.36	33.54
Multi-DecPrep-All	26.17	34.13
Multi-EncMerge-Bal	26.47	33.29
Multi-EncMerge-All	26.39	33.07
Multi-DecMerge-Bal	21.99	27.06
Multi-DecMerge-All	22.12	27.74
Specialized-Bal	27.43	34.32
Specialized-All	27.79	34.61

Table 5.2: BLEU on MuST-SHE.

languages. In particular, on the MuST-C test set, the scores of our ST Base models are 27.7 (en-it) and 40.3 (en-fr), respectively 0.3 and 4.8 BLEU above the best *cascade* results reported in (Bentivogli et al., 2020).

Moving on to ST systems, except for the Multi-DecMerge system (whose performance is significantly lower), we do not observe statistically significant BLEU differences between the Base models and their gender-aware extensions (Multi-* and Specialized-*), which also perform on par when fine-tuned with varying amounts of annotated data (balanced vs all).

Due to the very small percentage of speaker-dependent gender-marked words in MuST-SHE (< 3%, 810-840 over ∼30,000 words), the ability of the systems in translating gender is not reflected by BLUE scores. Therefore, we now delve deeper into our more informative evaluation (as per §5.3.5) and turn to the term coverage and gender accuracy values presented in Table 5.3. The overall results assessed with BLEU are confirmed by *term coverage* scores for both en-it and en-fr: the MT systems generate the highest number of annotated words present in MuST-SHE (63.83% on en-it and 63.10% on en-fr), while we do not observe large differences among the ST models (between 56.17% and 58.02% for en-it and 60.60% and 62.38%

	en-it		**en-fr**	
	Cover.	Acc.	Cover.	Acc.
MT for KD	63.83	51.45	63.10	52.08
Base	56.17	56.26	62.02	56.24
Multi-DecPrep-Bal	56.91	64.86	60.95	69.34
Multi-DecPrep-All	56.54	66.81	61.31	70.29
Multi-EncMerge-Bal	57.04	62.55	60.60	62.67
Multi-EncMerge-All	57.65	60.39	62.38	61.83
Multi-DecMerge-Bal	49.88	59.41	54.52	64.63
Multi-DecMerge-All	50.74	60.58	56.31	65.96
Specialized-Bal	57.90	86.35	61.79	86.13
Specialized-All	58.02	87.02	62.38	86.45

Table 5.3: Term coverage and gender accuracy.

for en-fr).

Instead, looking at *gender accuracy*, we immediately unveil that overall performance is not an indicator of the ability of systems in translating gender. In fact, the best-performing MT systems show the lowest gender accuracy (51.45% for en-it and 52.08% for en-fr): intrinsically constrained by the lack of access to audio information, they produce the wrong target gender in half of the cases. Base models indeed better translate gender, although they still present a large margin of improvement. Differently, the models fed with the speaker's gender information display a noticeable increase in gender translation, with Specialized-* models outperforming the Multi-* ones by 16–20 points and the Base ones by 30 points.

Among the multi-gender architectures, our results show that Multi-DecPrep has an edge on the other two models, both in overall and gender translation performance: for the sake of simplicity, from now on, we thus present only that model. As a single-model architecture, multi-gender would be a more functional solution than multiple specialized models, but – being trained on both female and male speakers' utterances – it is noticeably weaker than multiple specialized models (trained on gender-specific data) at predicting gender. With regard to the different amounts of gender-

	en-it				en-fr			
	Feminine		**Masculine**		**Feminine**		**Masculine**	
	Cover.	Acc.	Cover.	Acc.	Cover.	Acc.	Cover.	Acc.
MT for KD	66.25	16.23	61.46	88.49	63.76	16.24	62.41	89.58
Base	58.75	33.62	53.66	80.45	60.47	32.30	63.61	79.55
Multi-DecPrep-Bal	60.00	68.75	53.90	60.63	61.41	68.58	60.48	70.12
Multi-DecPrep-All	58.00	69.83	55.12	63.72	61.88	65.78	60.72	75.00
Specialized-Bal	62.00	79.84	53.90	93.67	62.59	79.32	60.96	93.28
Specialized-All	62.00	79.84	54.15	95.05	62.59	79.32	62.17	93.80

Table 5.4: Coverage and accuracy scores divided by feminine and masculine word forms.

annotated data used to train our gender-aware models, we cannot see any appreciable variation in term coverage and gender accuracy between the two settings.

Cross-gender Analysis

Table 5.4 shows separate term coverage and gender accuracy scores for target feminine and masculine forms. This allows us to highlight the translation ability of the models for each gender form and conduct cross-gender comparisons to detect potential bias. Also in this analysis, results are consistent across language pairs. We assess that the MT models present a very strong bias since they almost always produce masculine forms: accuracy is always much lower than 50% on the feminine set (up to 20.85% for en-it and 26.91% for en-fr) and very high on the masculine set (up to 88.49% for en-it and 89.58% for en-fr). The Base ST models improve the realization of feminine forms, but they still remain far from 50%.

All gender-aware models significantly reduce bias with respect to the Base systems. This is particularly evident in the feminine set, where accuracy scores far above 50% indicate their ability to correctly represent female speakers. In particular, the Specialized models achieve the best results on both feminine and masculine sets (over 79% and 93% respectively). The higher performance on the masculine set can be explained considering that the two gender-specialized models derive from the Base model, which is strongly biased towards masculine forms. Interestingly, Multi-DecPrep shows similar feminine/masculine accuracy scores. This is possibly due to the random initialization of the embeddings of gender tokens: as a result, the initial model hidden representations and predictions are perturbed in an unbiased way. An unbiased starting condition combined with balanced data leads to a fairer, similar behavior across genders, although the final models have lower accuracy than the Specialized ones.

Finally, we notice that results obtained by training our models with balanced (*-Bal) and unbalanced (*-All) datasets are similar. Indeed, the masculine gender accuracy slightly improves by adding more male data, while there is not a clear trend on the feminine accuracy: we can conclude that oversampling the data is functional inasmuch as it keeps the performance on the feminine set stable.

5.3.7 Analysis of Conflicting Vocal Characteristics and Tags

So far, we worked under the assumption that the speakers' vocal characteristics match those typically associated with the gender category they identify with. In this section, we explore the capacity of the systems in producing translations that are coherent with the speaker's gender in a scenario in which this assumption does not hold: this is the case of some transgenders, children and people with vocal impairment. However, we are hindered by the almost absent representation of such users within MuST-C. We hence design a counterfactual experiment where we associate the opposite gender tag to each actual female/male speaker and inspect the behavior of the models when receiving conflicting information between the gender tag and the properties of the acoustic signal.

Table 5.5 presents the results for this experiment. In the *M-audio/F-transl* set, systems were fed with a male voice and a female tag and the

	en-it				**en-fr**			
	M-audio/F-tag		**F-audio/M-tag**		**M-audio/F-tag**		**F-audio/M-tag**	
	Cover.	Acc.	Cover.	Acc.	Cover.	Acc.	Cover.	Acc.
Multi-DecPrep-All	54.88	45.78	60.25	38.17	61.93	45.14	61.18	55.77
Specialized-All	54.39	64.57	60.75	94.24	62.17	59.69	61.41	94.25

Table 5.5: Coverage and accuracy scores when the correct translation is expected in a gender form opposite to the speaker's gender but in accordance with the gender tag fed to the system.

expected translation is in the feminine form, while in the *F-audio/M-transl* set we have the opposite. As we can see, in both sets the multi-gender model has a drastic drop in accuracy with respect to the results shown in Table 5.4, with scores below 50% for en-it. This behavior indicates that the model relies on both the gender token and the audio features, which in this scenario are conflicting. Thus, the multi-gender model is not usable in scenarios in which the vocal characteristics have to be ignored. On the contrary, the specialized systems show a high accuracy on both sets. In particular, on *F-audio/M-transl* the performance is in line with the results of Table 5.4. This indicates that, independently of speakers' vocal characteristics, the model relies only on the provided gender information, being therefore suitable for situations in which one wants to control the gendered forms in the output and override the potentially misleading speech signals.

5.3.8 Summary

Going beyond the attested ability of direct ST systems in leveraging speakers' vocal characteristics from the audio input, we developed gender-aware models suitable for operating conditions where speakers' gender is known. To this aim, we annotated the MuST-C dataset with speakers' gender information, and used the new annotations to experiment with different architectural solutions: "multi-gender" and "specialized". The results of our experiments on two language pairs (en-it and en-fr) demonstrated the improvements in gender realization brought by breeding ST models aware of the speaker's gender. In particular, our specialized systems outperform the gender-unaware ST models by 30 points in gender accuracy without affecting overall translation quality. In addition, we demonstrated that specialized systems can be used to directly control the gender realization in the output even when the vocal characteristics of the speakers are conflicting with their gender, without requiring dedicated training data for this case. These solutions focused only on the gendered words referred to the speakers (category 1 of MuST-SHE). In the next sections, we turn our attention to the effect of well-established practices in ST on the translation of all gendered words (both category 1 and 2 of MuST-SHE).

5.4 The Effect of Word Segmentation

In line with the spirit of this chapter, which aims to investigate whether and which algorithmic aspects concur to exacerbate biased outputs, we now bring the analysis onto a seemingly neutral yet critical component: word segmentation. BPE represents the *de-facto* standard and has been recently shown to yield better results compared to character-based segmentation in ST (Di Gangi et al., 2020a). But does this hold true for gender translation as well? If not, why?

Languages like French and Italian often exhibit comparatively complex feminine forms, derived from the masculine ones by means of an additional suffix (e.g., en: *professor*, fr: *professeur* M vs. *professeure* F). Besides, women and their referential linguistic expressions of gender are typically under-represented in existing corpora (Hovy et al., 2020). In light of the above, purely statistical segmentation methods could be unfavorable for gender translation, as they can break the morphological structure of words and thus lose relevant linguistic information (Ataman et al., 2017). For instance, BPE merges the character sequences that co-occur more frequently, so rarer or more complex feminine-marked words may result in less compact sequences of tokens (e.g. en: *described*, it: *des@@critto* M vs. *des@@crit@@ta* F). Due to such typological and distributive conditions, may certain splitting methods render feminine gender less probable and hinder its prediction?

We address such questions by implementing different families of segmentation approaches employed on the decoder side of ST models built on the same training data. By comparing the resulting models both in terms of overall translation quality and gender accuracy, we explore whether a so-far considered irrelevant aspect like word segmentation can actually affect gender translation. As such, our contributions include: *i)* the first comprehensive analysis of the results obtained by 5 popular segmentation techniques for two language directions (en-fr and en-it) in ST; *ii)* finding that the target segmentation method is indeed an important factor and state-of-the-art subword splitting (i.e., BPE) comes at the cost of higher gender bias, while character-based models are the best at translating gender; *iii)* the proposal of a multi-decoder architecture able to combine BPE overall translation quality and the higher ability to translate gender of character-based segmentation.

5.4.1 Segmentation Techniques

For a comprehensive comparison of the impact of word segmentation on gender bias in ST, we identified three substantially different categories of splitting techniques. For each of them, we hereby present the candidates selected for our experiments.

Character Segmentation. Dissecting words at their maximal level of granularity, this technique proves simple and particularly effective at generalizing over unseen words. On the other hand, the length of the resulting sequences increases the memory footprint, and slows both the training and inference phases. We perform our segmentation by appending "@@ " to all characters but the last of each word.

Statistical Segmentation. This family comprises data-driven algorithms that generate statistically significant subwords units. The most popular is **BPE**,[11] which proceeds by merging the most frequently co-occurring characters or character sequences. Recently, He et al. (2020) introduced the Dynamic Programming Encoding **(DPE)** algorithm, which performs competitively and was claimed to accidentally produce more linguistically-plausible subwords with respect to BPE. DPE is obtained by training a mixed character-subword model. As such, the computational cost of a DPE-based ST model is around twice that of a BPE-based one. We trained the DPE segmentation on the transcripts and the target translations of the MuST-C training set, using the same settings of the original paper.[12]

Morphological Segmentation. A third possibility is linguistically-guided tokenization that follows morpheme boundaries. Among the unsupervised approaches, one of the most widespread tools is **Morfessor** (Creutz and Lagus, 2005), which was extended by Ataman et al. (2017) to control the size of the output vocabulary, giving birth to the **LMVR** segmentation method. These linguistically motivated segmentation techniques have outperformed other approaches when dealing with low-resource and/or morphologically-rich languages (Ataman and Federico, 2018). In other languages, they are

[11] We use SentencePiece implementation Kudo and Richardson (2018): `https://github.com/google/sentencepiece`.

[12] See `https://github.com/xlhex/dpe`.

	en-fr	**en-it**
# tokens	5.4M	4.6M
# types	96K	118K
BPE	8,048	8,064
Char	304	256
DPE	7,864	8,008
Morfessor	26,728	24,048
LMVR	21,632	19,264

Table 5.6: Dictionary sizes. “Tokens” refers to the number of words in the corpus, and not to the unit resulting from subword tokenization.

not as effective, so they are not widely adopted. Both Morfessor and LMVR have been trained on the MuST-C training set.[13]

For fair comparison, we chose the optimal vocabulary size for each method (when applicable). Following (Di Gangi et al., 2020a), we employed 8k merge rules for BPE and DPE, since the latter requires an initial BPE segmentation. In LMVR, instead, the desired target dimension is actually only an upper bound for the vocabulary size. We tested 32k and 16k, but we only report the results with 32k as it proved to be the best configuration both in terms of translation quality and gender accuracy. Finally, character-level segmentation and Morfessor do not allow determining the vocabulary size. Table 5.6 shows the size of the resulting dictionaries.

5.4.2 Experimental Settings

All the direct ST systems used in our experiments are built in the same fashion within a controlled environment, so to keep the effect of different word segmentations as the only variable. Accordingly, we train them on the MuST-C corpus, which contains 492 hours of speech for en-fr and 465 for en-it.

We are interested in measuring both *i)* the overall translation quality

[13] We used the parameters and commands suggested in `https://github.com/d-ataman/lmvr/blob/master/examples/example-train-segment.sh`

	en-fr			**en-it**		
	M-C	M-SHE	Avg.	M-C	M-SHE	Avg.
BPE	**30.7**	25.9	**28.3**	21.4	**21.8**	21.6
Char	29.5	24.2	26.9	21.3	20.7	21.0
DPE	29.8	25.3	27.6	21.9	21.7	**21.8**
Morfessor	29.7	25.7	27.7	21.7	21.4	21.6
LMVR	30.3	**26.0**	28.2	**22.0**	21.5	**21.8**

Table 5.7: SacreBLEU scores on MuST-C tst-COMMON (M-C) and MuST-SHE (M-SHE) for en-fr and en-it.

obtained by different segmentation techniques, and *ii)* the correct generation of gender forms. We evaluate translation quality on both the MuST-C *tst-COMMON* set and MuST-SHE, using SacreBLEU.[14] For fine-grained analysis on gender translation, we rely on gender accuracy (see §5.3.5). We report gender accuracy for the two categories of phenomena in MuST-SHE (see §5.2.1). In category 1, we let ST models leverage speakers' vocal characteristics as a gender cue to infer gender translation, since train and test data reflect this correlation.[15]

Concerning the architecture and training details, our models are Transformers adapted to ST, as in the previous section and chapters (see §3.3).

5.4.3 Comparison of Segmentation Methods

Table 5.7 shows the overall translation quality of ST systems trained with distinct segmentation techniques. BPE comes out as competitive as LMVR for both language pairs. On averaged results, it exhibits a small gap (0.2 BLEU) also with DPE on en-it, while it achieves the best performance on

[14] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.3.`

[15] Although potentially harmful for certain groups, we do not investigate methods to control gender translation, as we already did this in the previous section. Rather, we experimented with unmodified models for the sake of hypothesis testing without adding variability. However, our results suggest that, if certain word segmentation techniques better capture correlations from the received input, such capability could be exploited to redirect ST attention away from speakers' vocal characteristics by means of other information provided.

	ALL	1F	1M	2F	2M
			en-fr		
BPE	65.18	37.17	75.44	61.20	80.80
Char	**68.85**	48.21	74.78	65.89	**81.03**
DPE	68.55	**49.12**	70.29	**66.22**	80.90
Morfessor	67.05	42.73	75.11	63.02	80.98
LMVR	65.38	32.89	**76.96**	61.87	79.95
			en-it		
BPE	67.47	33.17	88.50	60.26	81.82
Char	**71.69**	**48.33**	85.07	**64.65**	**84.33**
DPE	68.86	44.83	81.58	59.32	82.62
Morfessor	65.46	36.61	81.04	56.94	79.61
LMVR	69.77	39.64	**89.00**	63.85	83.03

Table 5.8: Gender accuracy (%) for MuST-SHE Overall (ALL), Category 1 and 2 on en-fr and en-it.

en-fr. The disparities are small, though: they range within 0.5 BLEU, apart from Char standing ∼1 BLEU below. Compared to the scores reported by Di Gangi et al. (2020a), the Char gap is however smaller. As our results are considerably higher than theirs, we believe that the reason for such differences lies in a suboptimal fine-tuning of their hyperparameters. Overall, in light of the trade-off between computational cost (LMVR and DPE require a dedicated training phase for data segmentation) and average performance (BPE achieves winning scores on en-fr and competitive on en-it), we hold BPE as the best segmentation strategy in terms of general translation quality for direct ST.

Turning to gender translation, the gender accuracy scores presented in Table 5.8 exhibit that all ST models are clearly biased, with masculine forms (M) disproportionately produced across language pairs and categories. However, we intend to pinpoint the relative gains and losses among segmentation methods. Focusing on overall accuracy (ALL), we see that Char – despite its lowest performance in terms of BLEU score – emerges as the favorite segmentation for gender translation. For French, however, DPE is only slightly behind. Looking at morphological methods, they surprisingly do not outperform the statistical ones. The greatest variations are detected for feminine forms of Category 1 (1F), where none of the segmentation techniques reaches 50% of accuracy, meaning that they are all worse than a random choice when the speaker should be addressed by feminine expressions. Char appears close to such threshold, while the others (apart from DPE in French) are significantly lower.

These results illustrate that target segmentation is a relevant parameter for gender translation. In particular, they suggest that Char segmentation improves the model ability to learn correlations between the received input and gender forms in the reference translations. Although in this experiment models rely only on speakers' vocal characteristics to infer gender – which we discourage as a cue for gender translation for real-world deployment (see the previous section) – such ability shows a potential advantage for Char, which could be better redirected toward learning correlations with reliable gender meta-information included in the input. For instance, in a scenario in which meta-information (e.g., a gender tag) is added to the input to support gender translation, a Char model might better exploit this information. Lastly, our evaluation reveals that a proper comparison of gender translation potentialities of different solutions requires adopting the same segmentation. Our question then becomes: what makes Char segmentation less biased? What are the tokenization features determining a better/worse ability in generating the correct gender forms?

Lexical diversity. We posit that the limited generation of feminine forms can be framed as an issue of data sparsity, whereas the advantage of Char-based segmentation ensues from its ability to handle less frequent and unseen words (Belinkov et al., 2020). Accordingly, Vanmassenhove et al. (2018) and Roberts et al. (2020) link the loss of *linguistic diversity* (i.e.,

	en-fr		**en-it**	
	TTR	MATTR	TTR	MATTR
M-SHE Ref	*16.12*	*41.39*	*19.11*	*46.36*
BPE	14.53	39.69	17.46	44.86
Char	**14.97**	**40.60**	17.75	**45.65**
DPE	14.83	40.02	**18.07**	45.12
Morf	14.38	39.88	16.31	44.90
LMVR	13.87	39.98	16.33	44.71

Table 5.9: Lexical diversity scores on en-fr and en-it

the range of lexical items used in a text) with the overfitted distribution of masculine references in MT outputs.

To explore such hypothesis, we compare the lexical diversity (LD) of the translations produced by our models and MuST-SHE references. To this aim, we rely on Type/Token ratio (TTR – Chotlos 1944; Templin 1957), and the more robust Moving Average TTR (MATTR – Covington and McFall 2010).[16]

As we can see in Table 5.9, character-based models exhibit the highest LD (the only exception is DPE with the less reliable TTR metric on en-it). However, we cannot corroborate the hypothesis formulated in the above-cited studies, as LD scores do not strictly correlate with gender accuracy (Table 5.8). For instance, LMVR is the second-best in terms of gender accuracy on en-it, but shows a very low lexical diversity (the worst according to MATTR and second-worst according to TTR).

Sequence length. Italian and French feminine forms are, although to a different extent, longer and less frequent than their masculine counterparts. In light of such conditions, we expected that the statistically-driven BPE segmentation would leave feminine forms unmerged at a higher rate, and thus add uncertainty to their generation. To verify if this is the actual case –

[16] Metrics computed with software available at: `https://github.com/LSYS/LexicalRichness`. We set 1,000 as `window_size` for MATTR.

	en-fr (%)	**en-it** (%)
BPE	1.04	0.88
Char	1.37	0.38
DPE	2.11	0.77
Morfessor	1.62	0.45
LMVR	1.43	0.33

Table 5.10: Percentage increase of token-sequence length for feminine words over masculine ones.

explaining the lower gender accuracy of BPE models – we check whether the number of tokens (characters or subwords) of a segmented feminine word is higher than that of the corresponding masculine form. We exploit the coupled "wrong" and "correct" references available in MuST-SHE, and compute the average percentage of additional tokens found in the feminine segmented sentences[17] over the masculine ones. Results are reported in Table 5.10.

At first look, we observe opposite trends: BPE segmentation leads to the highest increment of tokens for feminine words in Italian, but to the lowest one in French. Also, DPE exhibits the highest increment in French, whereas it actually performs slightly better than Char on feminine gender translation (see Table 5.8). Hence, even the increase in sequence length does not seem to be an issue for gender translation. Nonetheless, these apparently contradictory results encourage our last exploration: *How* are gender forms actually split?

Gender isolation. By means of further manual analysis of 50 output sentences for each of the 6 systems, we inquire if longer token sequences for feminine words can be explained in light of the different characteristics and gender-productive mechanisms of the two target languages. Table 5.11

[17] As such references only vary for gender-marked words, we can isolate the difference relative to gender tokens.

	en	Segm.	F	M	Freq. F/M
a)	asked	BPE	chie–sta	chiesto	36/884
b)		DPE	chie–sta	chiesto	36/884
c)	friends	BPE	a–miche	amici	49/1094
d)		DPE	a–miche	amici	49/1094
e)	adopted	BPE	adop–tée	adop–té	30/103
f)		DPE	adop–t–é–e	adop–t–é	30/103
g)	sure	Morf.	si–cura	sicuro	258/818
h)	grown up	LMVR	cresci–uta	cresci–uto	229/272
i)	celebrated	LMVR	célébr–ées	célébr–és	3/7

Table 5.11: Examples of word segmentation. The segmentation boundary is identified by ”–”.

reports selected instances of coupled feminine/masculine segmented words, with their respective frequency in the MuST-C training set.

Starting with Italian, we find that BPE sequence length increment indeed ensues from greedy splitting that, as we can see from examples *(a)* and *(c)*, ignores meaningful affix boundaries for both same length and different-length gender pairs, respectively. Conversely, on the French set – with 95% of feminine words longer than their masculine counterparts – the low increment of BPE is precisely due to its loss of semantic units. For instance, as shown in *(e)*, BPE does not preserve the verb root (*adopt*), nor isolates the additional token (-*e*) responsible for the feminine form, thus resulting into two words with the same sequence length (2 tokens). Instead, DPE, which achieved the highest accuracy results for en-fr feminine translation (Table 5.8), treats the feminine additional character as a token *per se* (*f*).

Based on such patterns, our intuition is that the proper splitting of the morpheme-encoded gender information as a distinct token favors gender translation, as models learn to productively generalize it. Considering the high increment of DPE tokens for Italian in spite of the limited number of longer feminine forms (15%), our analysis confirms that DPE is unlikely to isolate gender morphemes on the en-it language pair. As a matter of fact, it produces the same kind of coarse splitting as BPE (see *(b)* and *(d))*.

Finally, we attest that the two morphological techniques are not equally valid. Morfessor occasionally generates morphologically incorrect subwords for feminine forms by breaking the word stem (see example *(g)* where the correct stem is *sicur*). Such behavior also explains the higher token increment of Morfessor with respect to LMVR. Instead, although LMVR (examples *(h)* and *(i)*) produces linguistically valid suffixes, it often condenses other grammatical categories (e.g., tense and number) with gender. As suggested above, if the pinpointed split of morpheme-encoded gender is a key factor for gender translation, the lower level of granularity of LMVR explains its reduced gender accuracy. Working on character sequences, instead, the isolation of the gender unit is always attained.

5.4.4 Beyond the Quality-Gender Trade-off

Informed by our experiments and analysis, we conclude this study by proposing a model that combines BPE overall translation quality and Char ability to translate gender. To this aim, we train a multi-decoder approach that exploits both segmentations to draw on their corresponding advantages.

In the context of ST, several multi-decoder architectures have been proposed, usually to jointly produce both transcripts and translations with a single model. Among those in which both decoders access the encoder output, here we consider the best-performing architectures according to Sperber et al. (2020). As such, we consider: *i) Multitask direct*, a model with one encoder and two decoders, both exclusively attending the encoder output as proposed by Weiss et al. (2017), and *ii)* the *Triangle* model (Anastasopoulos and Chiang, 2018), in which the second decoder attends the output of both the encoder and the first decoder.

For the triangle model, we used a first BPE-based decoder and a second Char-based decoder. With this order, we aimed to enrich BPE high-quality translation with a refinement for gender translation, performed by the Char-

	en-fr			**en-it**		
	M-C	M-SHE	Avg.	M-C	M-SHE	Avg.
BPE	**30.7**	25.9	28.3	21.4	21.8	21.6
Char	29.5	24.2	26.9	21.3	20.7	21.0
BPE&Char	30.4	**26.5**	**28.5**	**22.1**	**22.6**	**22.3**

Table 5.12: SacreBLEU scores on MuST-C tst-COMMON (M-C) and MuST-SHE (M-SHE) on en-fr and en-it.

	ALL	1F	1M	2F	2M
			en-fr		
BPE	65.18	37.17	**75.44**	61.20	80.80
Char	**68.85**	**48.21**	74.78	65.89	81.03
BPE&Char	68.04	40.61	75.11	**67.01**	**81.45**
			en-it		
BPE	67.47	33.17	**88.50**	60.26	81.82
Char	**71.69**	48.33	85.07	**64.65**	**84.33**
BPE&Char	70.05	**52.23**	84.19	59.60	81.37

Table 5.13: Gender accuracy (%) for MuST-SHE Overall (ALL), Category 1 and 2 on en-fr and en-it.

based decoder. However, the results were negative: the second decoder seems to excessively rely on the output of the first one, thus suffering from a severe *exposure bias* (Ranzato et al., 2016) at inference time. Hence, we do not report the results of these experiments.

Instead, the *Multitask direct* has one BPE-based and one Char-based decoder. The system requires a training time increase of only 10% and 20% compared to, respectively, Char and BPE models, while, during inference, running time and size are the same as a BPE model. We report overall translation quality (Table 5.12) and gender accuracy (Table 5.13) of the BPE output (BPE&Char).[18] Starting with gender accuracy, the overall gender translation ability (ALL) of the Multitask model is still lower, although very close, to that of the Char-based model. Nevertheless, feminine translation improvements are present on Category 2F for en-fr and, with a larger gain,

[18] The Char scores are not reported, as they are not enhanced compared to the base Char encoder-decoder model.

on 1F for en-it. We believe that the presence of the Char-based decoder is beneficial to capture into the encoder output gender information, which is then also exploited by the BPE-based decoder. As the encoder outputs are richer, overall translation quality is also slightly improved (Table 5.12). This finding is in line with other work (Costa-jussà et al., 2020), which proved a strict relation between gender accuracy and the amount of gender information retained in the intermediate representations (encoder outputs).

Overall, following these considerations, we posit that target segmentation can directly influence the gender information captured in the encoder output. In fact, since the Char and BPE decoders do not interact with each other in the Multitask model, the gender accuracy gains of the BPE decoder cannot be attributed to a better ability of a segmentation method in rendering the gender information present in the encoder output into the translation. With this work, we have taken a step forward in ST for English-French and English-Italian, pointing at plenty of new ground to cover concerning *how to* split for different language typologies.

5.4.5 Summary

After the study on how to control the gender of words referring to the speaker (§5.3), in this section we continued our exploration of the exacerbation of gender bias caused by technical choices in direct ST models, focusing on the influence of word segmentation. To this aim, we compared several word segmentation approaches on the target side of ST systems for English-French and English-Italian, in light of the linguistic gender features of the two target languages. Our results show that word segmentation does affect gender translation and that the higher BLEU scores of state-of-the-art BPE-based models come at the cost of lower gender accuracy. Moreover, our analyses of the behavior of different segmentation techniques revealed that improved generation of gender forms could be linked to the proper isolation of the morpheme that encodes gender information, a feature that is attained by character-level segmentation. Lastly, we introduced a multi-decoder training strategy to leverage the qualities of BPE and character splitting, improving both gender accuracy and BLEU score, while keeping computational costs under control. The next section enriches the comparison between BPE-based and char-based models by introducing new fine-grained annotations of the gender phenomena in MuST-SHE, and assessing their ability in handling different part-of-speech and morphosyntactic agreement chains.

5.5 A Multifaceted Evaluation

The previous sections (§5.3 and §5.4) and previous works (see §5.2.1) assessed the bias of ST systems with word-level metrics that treat all gender-marked words indiscriminately. Indeed, the existing benchmarks do not allow us to inspect if and to what extent different word categories (or part-of-speech) participate in gender bias and overlook the underlying morphosyntactic nature of grammatical gender on agreement chains, which cannot be monitored on single isolated words (e.g., *en*: a strange friend; *it*: una/o strana/o amica/o).[19]

We believe that fine-grained evaluations including the analysis of gender agreement across different parts of speech (POS) are relevant not only to gain a deeper understanding of bias in grammatical gender languages, but also to inform mitigation strategies and data curation procedures. Toward these goals, our contributions[20] are as follows: *i)* we enrich MuST-SHE with two layers of linguistic information, POS and agreement chains;[21] *ii)* in

[19] To be grammatically correct, each word in the chain has to be inflected with the same (masculine or feminine) gender form, similar to number agreement.

[20] The creation of the annotation layer has been curated by Beatrice Savoldi and Luisa Bentivogli. However, it is reported here as it is crucial for the comprehension of the other contributions.

[21] The annotation layers are an extension of MuST-SHE v1.2 and are freely downloadable at:

light of the findings of the previous section, we rely on our manually curated resource to compare three ST models, which are trained on varying amounts of data, and built with different segmentation techniques (character and BPE). Lastly, through experiments on three language pairs (en-es, en-fr, en-it) we demonstrate that *iii)* not all POS are equally impacted by gender bias, and *iv)* translating words in agreement does not emerge as a systematic issue.

5.5.1 MuST-SHE Enrichment

In light of the above, a fine-grained evaluation of bias focused on POS and gender agreement requires the creation of a new dedicated resource. Rather than building it from scratch, we add two annotation layers to the existing MuST-SHE benchmark.[22] The target languages covered in MuST-SHE (es, fr, it) are particularly suitable to focus on linguistic specificity. In fact, as Gygax et al. (2019) suggest, accounting for gender in languages with similar typological features allows for proper comparison.

Phenomena Categorization

Parts-Of-Speech. We annotate each target gender-marked word in MuST-SHE with POS information. As shown in Table 5.14 (*a-c*), we differentiate among six POS categories:[23] *i)* articles, *ii)* pronouns, *iii)* nouns, *iv)* verbs. For adjectives, we further distinguish *v)* limiting adjectives with minor semantic import that determine e.g., possession, quantity, space (*my*, *some*, *this*); and *vi)* descriptive adjectives that convey attributes and qualities, e.g. *glad*, *exhausted.* This distinction enables to neatly sort our POS categories into the closed class of function words, or into the open one of content

ict.fbk.eu/must-she/ under the same MuST-SHE licence (CC BY NC ND 4.0)

[22] Version 1.2: `https://ict.fbk.eu/must-she/`

[23] Some POS categories (e.g., conjunctions, adverbs) are not considered since they are not subject to gender inflection.

		PARTS-OF-SPEECH
(a)	SRC	As *one* of the *first* women...
	REF$_{fr}$	En tant que l'**une**$_{Pron}$ des **premières**$_{Adj-det}$ femmes..
(b)	SRC	As a *child growing up* in Nigeria...
	REF$_{it}$	Da **bambino**$_{Noun}$ **cresciuto**$_{Verb}$ in Nigeria.
(c)	SRC	Then *an amazing* colleague...
	REF$_{es}$	Luego **una**$_{Art}$ **asombrosa**$_{Adj-des}$ colega...
		AGREEMENT
(d)	SRC	I was *the first Muslim* homecoming queen, *the first* Somali student *senator...*
	REF$_{es}$	Fui [**la primera** reina **musulmana**] del baile, [**la primera senadora**] somalí estudiantil...
(e)	SRC	She's also *been interested* in research.
	REF$_{it}$	E' [**stata** anche **attratta**] dalla ricerca .
(f)	SRC	I also *became a* high school *teacher.*
	REF$_{fr}$	Je suis aussi [**devenu un professeur**] de lycée.

Table 5.14: MuST-SHE target **gender-marked words** annotated per $_{POS}$ and [agreement chains].

words (Schachter and Shopen, 2007). Since words from these two classes differ substantially in terms of variability, frequency, and semantics, we reckon they represent a relevant variable to account for in the evaluation of gender bias.

Agreement. We also enrich MuST-SHE with linguistic information that is relevant to investigate the morphosyntactic nature of grammatical gender agreement. Gender agreement, or *concord* (Corbett, 2006; Comrie, 1999), requires that related words match the same gender form, as in the case of *phrases*, i.e. groups of words that constitute a single linguistic unit.[24] Thus, as shown in Table 5.14, we identify and annotate as agreement chains gender-marked words that constitute a phrase, such as a noun plus its modifiers (d), and verb phrases for compound tenses (e). Also, structures that involve a gender-marked (semi-) copula verb and its predicative complement are

[24] If agreement is not respected, the unit becomes ungrammatical e.g. *es*: *el$_M$ buen$_M$ niña$_F$ (the good kid).

annotated as chains (*f*), although in such cases the agreement constraint is "weaker".[25] This annotation lets us verify whether a model consistently picks the same gender paradigm for all words in the chain, enabling the assessment of its syntagmatic behavior.

Manual annotation

POS and agreement annotation was manually carried out by 6 annotators (2 per language pair) undergoing a linguistics/translation studies MA degree, and with native/excellent proficiency in the assigned target language. For each language pair, they annotated the whole corpus independently, based on detailed guidelines.[26] For POS, we computed inter-annotator agreement (IAA) on label assignment with the kappa coefficient (in Scott's π formulation – Scott 1955). The resulting values of 0.92 (en-es), 0.94 (en-fr) and 0.96 (en-it) correspond to "almost perfect" agreement according to its standard interpretation (Landis and Koch, 1977). For gender agreement, IAA was calculated on the exact match of the complete chains in the two annotations. The resulting Dice coefficients (Dice, 1945) of 89.23% (en-es), 93.0% (en-fr), and 94.34% (en-it) can be considered highly satisfactory given the more complex nature of this latter task. Except for few liminal cases that were excluded from the dataset, all disagreements were reconciled.

We show the final annotation statistics in Table 5.15. Variations across languages are due to inherently cross-lingual differences.[27] These figures underscore the so far largely unaccounted variability of gender across lexical categories.

[25] Such structure, due to the semantics of some linking verbs, can enable more flexibility. E.g., in French, *Elle est devenue*$_F$ *un*$_M$ *canard*$_M$ (*She became a duck*) is grammatical, although *un canard* (a duck) is formally masculine.

[26] The full guidelines are available ar: https://bit.ly/3CdU50s.

[27] Spanish, for instance, relies less than French or Italian on the gender-enforcing to be auxiliary, resulting in less gender-marked verbs (*fr*: est parti/ie; *it*: è partita/o; *es*: se ha ido).

	en-es	en-fr	en-it	M-SHE All
POS (tot)	2099	1906	2026	6031
Art	487	325	413	1225
Pronoun	104	61	48	213
Adj-det	118	106	149	373
Adj-des	676	576	448	1700
Noun	607	344	346	1297
Verb	107	494	622	1223
AGR-CHAINS	420	293	421	1080

Table 5.15: Distribution of POS and agreement chains for each language and in the whole MuST-SHE corpus.

5.5.2 Experimental Settings

Our experiments draw on the finding of the previous section and of studies exploring the relation between overall system performance, model size and gender bias. Vig et al. (2020) posit that bias increases with model size as larger systems better emulate biased training data. Working on WinoMT/ST, (Kocmi et al., 2020) correlates higher BLEU scores and gender stereotyping, whereas (Costa-jussà et al., 2022) shows that systems with lower performance tend to produce fewer feminine translations for occupations, but rely less on stereotypical cues. To account for these findings and inspect the behavior of different models under natural conditions, we experiment on three language pairs (en-fr, en-it, en-es) with three direct ST solutions, namely: LARGE-BPE, SMALL-BPE, and SMALL-CHAR.

Our SMALL-BPE and SMALL-CHAR models are the same of the previous section. The LARGE-BPE systems are also analogous, but their training is performed in three consecutive steps on both ST corpora and synthetic data obtained by automatically translating the ASR corpora transcript (see §3.3). The dataset used are the same of §3.3 as well. Trainings are performed on 4 GPUs and stopped after 5 epochs without improvements on the validation loss. We average 5 checkpoints around the best on the validation set. As a validation set, we rely on the MuST-C gender-balanced

	en-es	en-fr	en-it
(Bentivogli et al., 2021)	32.93	-	28.56
(Le et al., 2021)	28.73	34.98	24.96
LARGE-BPE	34.1	40.3	27.7

Table 5.16: Comparison in terms of BLEU scores of LARGE-BPE models with recent works.

dev set, introduced in §5.3.3.

We employ the enriched MuST-SHE corpus to assess generic performance and gender translation at several levels of granularity. We measure translation quality with SacreBLEU,[28] while for *word-level* gender-specific evaluations we rely on the coverage, and gender accuracy metrics introduced in §5.3.5. For *chain-level* gender agreement evaluation, we define coverage as the proportion of fully generated chains. Then, we adapt gender accuracy[29] to measure the proportion of fully generated agreement chains for: *i)* agreement-correct, i.e. agreement is respected and with the correct gender; *(ii)* agreement-wrong, i.e. agreement is respected, but with the wrong gender form; and *(iii)* no-agreement, i.e. agreement is not respected, as both feminine and masculine gender inflections occur within the same chain.

5.5.3 Results

Overall Quality and Gender Translation

First, to ensure the trustworthiness of our results, we compare our LARGE-BPE with recently published results on MuST-C test data. Table 5.16 shows that our systems compare favorably overall on the three language pairs.

Then, we turn to compare the overall and gender translation quality of our systems. Table 5.17 reports the results that exhibit a consistent trend over the three language directions: unsurprisingly, LARGE-BPE systems

[28] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.3`

[29] The scripts has been released together with the extensions.

		BLEU	**All-Cov**	**All-Acc**	**F-Acc**	**M-Acc**
en-es	SMALL-BPE	27.6	65.0	64.1	45.8	79.6
	SMALL-CHAR	26.5	64.2	67.3	**52.8**	79.6
	LARGE-BPE	**34.1**	**72.0**	**69.1**	**52.8**	**83.6**
en-fr	SMALL-BPE	25.9	55.7	64.9	50.3	78.1
	SMALL-CHAR	24.2	55.9	68.5	**57.7**	78.2
	LARGE-BPE	**34.3**	**64.3**	**70.9**	57.1	**83.4**
en-it	SMALL-BPE	21.0	53.1	67.7	52.3	80.3
	SMALL-CHAR	20.7	52.6	**71.6**	**57.2**	83.9
	LARGE-BPE	**27.5**	**59.2**	69.1	52.2	**85.4**

Table 5.17: BLEU, coverage, and gender accuracy scores computed on MuST-SHE.

achieve by far the highest overall translation quality, while SMALL-BPE models outperform the CHAR ones as shown in the previous section. The higher overall translation quality of LARGE-BPE models is also reflected by the coverage scores (All-Cov), indicating that they generate the highest number of MuST-SHE gender-marked words for all language pairs.

By turning to overall gender accuracy (All-Acc) though, the edge previously assessed for the bigger state-of-the-art systems ceases to be clear-cut: for en-es and en-fr LARGE-BPE systems outperform SMALL-CHAR by ∼2 points only, a slim advantage compared to the huge gap observed in BLEU score, whereas SMALL-CHAR proves the best at translating gender for en-it. We further zoom into the comparison of gender translation for feminine (F-Acc) and masculine (M-Acc) forms, where we can immediately assess that all ST models are skewed toward a disproportionate production of masculine forms (on average, 53.1% for F vs. 81.3% for M). Focusing on LARGE-BPE models, we discover that their higher global gender accuracy (All-Acc) is actually due to the higher generation of masculine forms, while they do not compare favorably when it comes to feminine translation. In fact, in spite of achieving the lowest generic translation quality, SMALL-CHAR proves on par (for en-es) or even better (for en-it and en-fr) than LARGE-BPE at handling feminine gender translation.

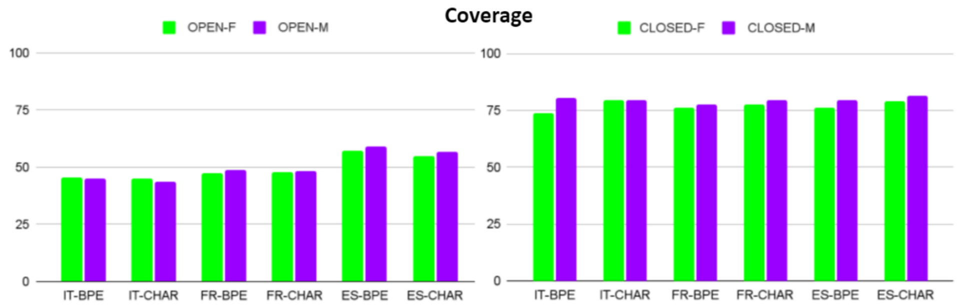


Figure 5.1: F *vs* M coverage per open and closed class words.

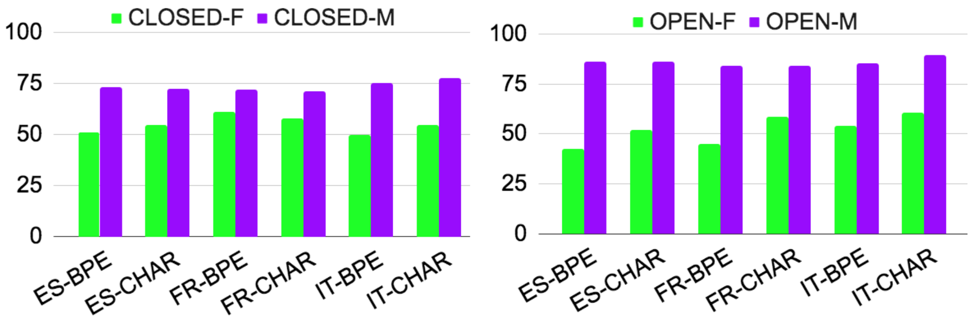


Figure 5.2: F *vs.* M accuracy for closed and open class words.

In light of the above, our results reiterate the importance of dedicated evaluations that, unlike holistic metrics, are able to disentangle gender phenomena. As such, we can confirm that higher generic performance does not entail a superior capacity in producing feminine gender. This does not only emerge in the comparison of (small) BPE- and char-based ST models, as shown in the previous section. Rather, even for stronger systems, we attest how profiting from a wealth of – uncurated and synthetic (Bender et al., 2021) – data does not grant advantages to address gender bias. This motivates us to continue our multifaceted evaluation by taking into account only small models – henceforth CHAR and BPE – that, being trained on the same MuST-C data, allow for sound and transparent comparison.

		Verbs		**Nouns**		**Adj-des**	
		F-Acc	M-Acc	F-Acc	M-Acc	F-Acc	M-Acc
en-es	BPE	44.4	**93.8**	21.1	89.0	57.4	**80.0**
	CHAR	**60.0**	84.2	**37.4**	**89.7**	**61.2**	79.7
en-fr	BPE	51.3	**79.8**	16.4	93.5	50.6	78.6
	CHAR	**68.4**	75.0	**27.4**	**95.3**	**63.0**	**81.4**
en-it	BPE	63.7	83.7	28.6	92.2	62.0	76.7
	CHAR	**66.7**	**89.2**	**33.3**	**94.3**	**70.6**	**84.5**

Table 5.18: F *vs.* M Accuracy scores per open class POS.

Word Classes and Parts-of-speech

At a finer level of granularity, we use the MuST-SHE extension to inspect gender bias across open and closed class words. Their coverage ranges between 74-81% for function words, but it shrinks to 44-59% for content words (see Figure 5.1). This is expected given the limited variability and high frequency of functional items in language. Instead, the coverage of feminine and masculine forms is on par within each class for all systems, thus allowing us to evaluate gender accuracy on a comparable proportion of generated words. A bird's-eye view of Figure 5.2 attests that, although masculine forms are always disproportionately produced, the gender accuracy gap is amplified on the open class words. The consistency of such behavior across languages and systems suggests that content words are involved to a greater extent in gender bias.

We hence analyze this more problematic class by looking into a breakdown of the results per POS. Table 5.18 presents results for *verbs*, *nouns*, and *descriptive adjectives*. First, in terms of system capability, CHAR models still consistently emerge as the best for feminine translation. What we find notable, though, is that even within the same class we observe evident differences, where nouns come forth as the most biased POS with a huge divide between M and F accuracy (52–77 points). Specifically, scores below 50% indicate that feminine forms are generated with a probability that is below random choice, thus signaling an extremely strong bias.

		Art		**Pronoun**		**Adj-det**	
		F-Acc	M-Acc	F-Acc	M-Acc	F-Acc	M-Acc
en-es	*bpe*	51.35	**70.0**	**52.0**	84.9	49.1	86.1
	char	**53.5**	68.4	51.7	**85.7**	**59.3**	**91.2**
en-fr	*bpe*	**52.0**	**69.2**	**65.5**	**78.3**	**82.9**	**79.5**
	char	50.8	68.6	54.2	77.3	79.1	78.6
en-it	*bpe*	47.2	74.6	**75.0**	71.4	50.9	81.8
	char	**52.2**	**76.8**	52.9	**77.8**	**61.8**	**83.3**

Table 5.19: F *vs.* M accuracy scores per closed class POS.

In light of this finding, we hypothesize that semantic and distributional features might be a factor to interpret the word gender skew. Specifically, occupational lexicon (e.g., lawyer, professor) makes up for most of the nouns represented in MuST-SHE (∼70%). While such a high rate of professions in TED data is not surprising *per se*,[30] it singles out that professions may actually represent a category where systems largely rely on spurious cues to perform gender translation, even within natural conditions that do not ambiguously prompt stereotyping. We exclude basic token frequency by POS as a key factor to interpret our results, as MuST-SHE feminine nouns do not consistently appear as the POS with the lowest number of occurrences, nor do they have the lowest F:M ratio within MuST-C training data.

We conclude our evaluation of gender accuracy of different POS by looking at the function words. As we can see in Table 5.19, the otherwise attested advantage of CHAR over BPE is not consistent for function words, where we find variations across POS and languages. Such variations may be due to the fairly restricted amount of MuST-SHE *pronouns* and *limiting adjectives* (Adj-det) on which accuracy can be computed in MuST-SHE (see Table 5.15), which make very fine-grained evaluations particularly unstable.

[30] As TED talks are held by field experts, references to education and titles are quite common (MacKrill et al., 2021).

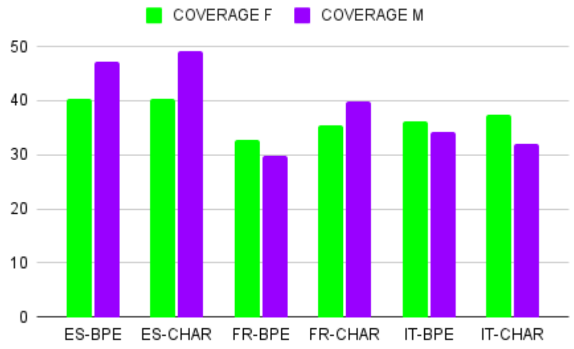


Figure 5.3: F *vs* M chains coverage

Additionally – since the present POS evaluation still remains at the word level – we are not able to ponder whether gender translations for modifiers (i.e., articles, determiners) is to some extent constrained by the content words they refer to.

Gender Agreement Evaluation

The final step in our multifaceted analysis goes beyond the word level to inspect agreement chains in translation. Overall, agreement translation was measured on a lower *coverage* (30-50%) than the world-level one (see Table §5.17) as expected given the strict requirement of generating full chains with several words. Figure 5.3 shows the coverage of fully generated agreement chains split into feminine (F) and masculine (M) forms. Although we attest notable variations across languages and gender forms, overall masculine and feminine chains are both produced at comparable rates.

Table 5.20 shows accuracy scores for all MuST-SHE agreement chains (All), also split into feminine (F) and masculine (M) chains. The overall results are promising: we find very few instances (literally 1 or 2) in which ST systems produce an ungrammatical output that breaks gender agreement (NO). In fact, both systems tend to be consistent with one picked gender for the whole dependency group. Thus, in spite of previous MT studies

		All			**Feminine**			**Masculine**		
		C	W	NO	C	W	NO	C	W	NO
en-es	*bpe*	74.3	**24.6**	**1.2**	33.9	**64.4**	**1.7**	95.5	**3.6**	0.9
	char	**78.4**	21.0	0.6	**42.4**	57.6	0.0	**96.6**	2.6	0.9
en-fr	*bpe*	67.9	**31.0**	**1.2**	54.1	**45.9**	0.0	78.7	**19.1**	**2.1**
	char	**76.7**	22.3	1.0	**57.5**	40.0	**2.5**	**88.9**	11.1	0.0
en-it	*bpe*	71.7	**27.5**	0.7	47.4	**50.9**	**1.8**	88.9	**11.1**	0.0
	char	**78.5**	20.0	**1.5**	**54.2**	44.1	1.7	**97.4**	1.3	**1.3**

Table 5.20: Accuracy scores for gender agreement. Scores are given for agreement respected with correct gender (C), agreement respected with wrong gender (W), and agreement not respected (NO).

concluding that character-based segmentation results in poorer syntactic capability (Belinkov et al., 2020), respecting concord does not appear as an issue for any of our small ST models. For the sake of comparability, however, we note that our evaluation involves language pairs that do not widely resort to long-range dependencies; this may contribute to explaining why CHAR better handles correct gender agreement.

5.5.4 Summary

Following the findings of the previous section that unveiled the importance of the target text segmentation method, in this section we explored whether different POS categories are equally suject to gender bias and whether grammatical gender agreement is respected in the output of ST models built with different segmentation techniques and data quantities. To this aim, we enriched the MuST-SHE benchmark with new linguistic information and carried out an extensive evaluation on 3 language pairs (en-es/fr/it), which led to two main findings. First, while all POS categories are subject to masculine skews, they are not impacted to the same extent, as nouns represent the category that is mostly affected by bias. Second, respecting gender agreement in the translation of related words is not an issue for current ST models. In line with the previous section, we also reiterated that, in spite of lower generic performance, character-based segmentation favors feminine translation at different levels of granularity. In addition, we demonstrated that translation quality and the amount of training data do not influence how much models are biased. With the same spirit of understanding how techniques introduced to improve the quality of ST systems affect their gender bias, we now turn to analyze the effects on gender bias of distilling knowledge from an MT teacher.

5.6 Knowledge Distillation and Gender Translation

We conclude our investigation on the impact of training and architectural solutions on gender bias by complementing the study on KD for ST presented in §3.3. As we have seen, distilling knowledge from an MT system improves the translation quality of ST students and this technique is widely employed to train strong ST models (Zhang and Ao, 2022; Zhang et al., 2022). However, the ST input (audio) contains information that is not present in the MT input (the corresponding transcript). As an example, the sentence *"I am a student"* can be translated into Italian either as *"Sono uno studente"* or as *"Sono una studentessa"* depending on the gender of the speaker. As this information is completely missing in the textual English input, in section §5.3 we have seen that an MT model is likely to produce the more frequent masculine forms with representational harm for women. The speaker's pitch in the speech input, instead, can be used as a gender cue to disambiguate the correct form. Although in general biological features should not be considered as gender cues,[31] our training dataset (MuST-C) contains a strong correlation between speakers' vocal characteristics and

[31] The adoption of physical cues can lead to reductionist gender classifications (Zimman, 2020) and be harmful to a diverse range of users.

gender forms in the reference translations, so ST models can learn and leverage this gender cue in our setting. Our question therefore is: do ST students also learn the bias in MT models as a side effect?

To answer this question, we analyzed the behavior of ST models trained with KD on the category 1 of the MuST-SHE English→{French, Italian} sections. Our investigation led to the following findings: *i)* ST students learn not only useful information, but also the gender bias in MT models, in particular for gender-marked words that are related to the speaker; *ii)* the fine-tuning on ST corpora introduced in §3.3 eliminates this additional gender bias.

5.6.1 Results

To assess the gender bias of models trained with KD from an MT teacher, we inspect the behavior of the systems trained in high resource conditions described in §3.3. These systems undergo a first training with `Word-KD` on ASR data augmented with a pseudo-reference translation generated by the MT teacher (`Seq-KD`). Then, they are fine-tuned with `Word-KD` on ST corpora, before a final fine-tuning on ST corpora without KD, in accordance with the findings of §3.3. For full experimental settings and architectural details, the reader can refer to §3.3.2.

As a baseline, we report two systems trained without KD: the ST system developed by Bentivogli et al. (2020), where the target text is represented at character level (`Base Char ST`), and the BPE-based system presented in §5.4 (`Base BPE ST`). Indeed, we demonstrated in §5.4 and §5.5 that target-text segmentation is an important factor for the system ability to translate gender and our systems segment target text with BPE, as this text segmentation method leads to the best translation quality. We measure the ability in translating gender with gender accuracy (see §5.3.5).

Since we are interested in assessing the effect of KD on the ability of the

	BLEU	**Female Gender Acc.**	**Male Gender Acc.**
	en-it		
Base Char ST	21.5	**49.5%**	87.2%
Base BPE ST	21.8	33.2%	**88.5%**
MT	**33.6**	16.3%	**88.5%**
`Seq-KD` + `Word-KD` + FT `Word-KD`	23.6	20.9%	84.9%
+ FT w/o KD	27.5	33.6%	80.5%
	en-fr		
Base Char ST	27.9	**46.3%**	86.2%
Base BPE ST	25.9	37.2%	75.4%
MT	**39.6**	16.2%	**89.6%**
`Seq-KD` + `Word-KD` + FT `Word-KD`	32.0	26.9%	79.4%
+ FT w/o KD	34.3	32.3%	79.6%

Table 5.21: BLEU score and Gender Accuracy on Category 1F (female speakers) and 1M (male speakers) of the MuST-SHE test set.

resulting ST systems to deal with gender, we compare: *i)* the teacher MT models, *ii)* the intermediate ST models trained on KD, and *iii)* the final ST models obtained with fine-tuning without KD. The results are reported in Table 5.21. First, we confirm that overall performance is not an indicator of the system ability to translate gender. In fact, the best-performing MT systems show the lowest female gender accuracy. Such deficiency is directly reflected in the ST students (`Seq-KD` + `Word-KD` + FT `Word-KD`), which are strongly influenced by the MT behavior; thus, although effective for overall quality, KD is detrimental to gender translation. However, fine-tuning on ST data demonstrates beneficial also by improving gender accuracy of the feminine forms from 20.9-26.9% to 33.6-32.6% respectively on en-it and en-fr, reducing the bias towards generating masculine forms. In particular, the gap with a BPE-based ST system (`Base BPE ST`) is closed (en-it – 33.6% vs 33.2%) or significantly reduced (en-fr – 32.3% vs 37.2%). So, the fine-tuning seems to completely remove the additional bias of the ST student compared to a normal ST system. The gap with the ST systems by Bentivogli et al. (2020) is, instead, still large (33.6% vs 49.5% on en-it, 32.3% vs 46.5% on en-fr), but it is motivated by the different text segmentation (char vs BPE).

All in all, the experiments show that distilling knowledge from biased MT models is detrimental when it comes to gender bias. However, the final fine-tuning without KD mitigates the additional gender bias and the resulting models display a similar bias to systems trained from scratch.

5.6.2 Summary

This section concludes our investigation on the impact on gender bias of technical solutions and choices motivated by the pursuit of higher translation quality. In particular, we complemented and completed our in-depth study on the distillation of knowledge from an MT teacher (§3.3), by assessing its effect on the bias of the ST student. Through experiments on two language pairs, we demonstrated that: *i)* KD introduces additional bias in the ST systems, as words referring to the speaker are almost always realized in their masculine forms; *ii)* the problem can be solved by means of a fine-tuning on ST corpora.

5.7 Conclusions

The huge improvements in translation quality of direct ST systems described in the first two chapters do not imply that these models are ready to be useful for a wide range of users. Indeed, techniques that are effective in producing overall performance gains may be detrimental for certain groups, hindering their access to this new technology. As such, in this chapter we studied different techniques that are widely adopted in ST in light of their translation-quality benefit, showing that they lead to an increased gender bias and exacerbate the detraction from the representation of certain groups. For each of the issues detected, we then proposed solutions that go beyond the compromise between translation quality and gender bias. Specifically, we first demonstrated that, while direct ST models have an edge on cascade systems in translating feminine words referred to the speaker, there is still large room for improvement, and their reliance on biometrical cues makes them unsuitable for certain groups (e.g., transgender, children). To avoid these limitations, we *i)* released a new annotation of the TED talks in MuST-C and MuST-SHE with the speakers' gender, *ii)* presented a more informative evaluation procedure (which disentangles coverage and gender accuracy), and *iii)* showed that specialized models (fine-tuned for each gender category) allow for controlling the gender realization of speaker-referred words, even if they contrast with the biometrical cues.

Then, we unveiled that the higher translation quality brought by a segmentation of the target text with BPE comes at the cost of increasing gender bias with respect to a character-based segmentation, which emerged as the least biased text segmentation method even in comparison with morphologically-motivated splits. As a solution, we suggested a multi-decoder training strategy, in which a character-based decoder leads to more informative encoder outputs that are exploited by the BPE decoder, capable of achieving the quality of a pure BPE-based model and the gender accuracy of a character-based system.

In addition, we released new annotation layers over the MuST-SHE test set, allowing for a more fine-grained analysis of gender bias in ST systems. This resource enabled us to demonstrate that nouns exhibit the highest degree of bias among all POS, while gender agreement chains are respected by ST systems, and that their performance on these aspects does not depend on the translation quality and amount of training data of ST system, but rather by the specific techniques employed (e.g., the target text segmentation strategy).

At last, we completed the study of KD in §3.3, assessing its effect on gender bias. ST students came out as "good learners" also of the bias of

MT teachers, but a fine-tuning on ST corpora eliminated the additional bias introduced.

The next chapter concludes our investigation of the capabilities of direct ST systems by studying their application in the context of “augmented ST”, where the focus is mostly on the ability to correctly render and recognize named entities and specific terminology.

Chapter 6

Augmented Speech Translation

6.1 Introduction

In line with the spirit of Chapter 5, in this chapter we go beyond the good scores obtained in Chapter 3 and 4 for overall translation quality and assess specific aspects that – although crucial in real applications – are neglected by holistic measures. In particular, we focus on named entities (NEs) and terminology, whose correct handling is needed to convey the proper meaning of a sentence (Li et al., 2013) as translation errors often result in blatant (meaningless, hilarious, or even offensive) errors, which jeopardize users' trust in the translation system. One example is "moby dick" (in lower case, as in the typical output of a speech recognition system): Google Translate[1] returns *mazikó poulí* (massive bird) for Greek, while the translation contains profanities for other languages like Hungarian or Italian.

Regardless of their relevance, as anticipated in §1.2, automatic metrics are relatively insensitive to errors on NEs and numbers (Amrhein and Sennrich, 2022), which are instead of paramount importance for human readers (Xie et al., 2022). Specifically, in a human-centric vision of automatic translation, where technological solutions "augment" users by supporting and relieving them from the most tedious and cognitive-intensive tasks, translation tools

[1] Accessed on the 27th April 2021.

should primarily highlight specific terms and NEs, eventually enriching them with relevant/contextual information (Lommel, 2018). In this vision, translators and interpreters can dedicate themselves to crafting fluent and intelligible translations, a task they can easily address, differently from machines (Fantinuoli and Prandi, 2021), while machines carry out repetitive tasks, such as the identification, lookup into dictionaries, and disambiguation of domain-specific terminology and NEs, which significantly contribute to humans' high cognitive workload (Jones, 1998; Prandi, 2018; Desmet et al., 2018). Indeed, NEs and terminology *i)* are hard to remember for interpreters (Liu et al., 2004), *ii)* can be unknown to interpreters/translators and difficult to recognize (Griffin and Bock, 1998), and *iii)* differently from other types of words, usually have one or few correct translations. For this reason, modern computer-assisted interpreting (CAI – Fantinuoli 2017) tools aim at automatically recognizing, displaying, and translating NEs and terms. However, current solutions rely on pre-defined dictionaries to identify and translate the elements of interest (Fantinuoli et al., 2022), preventing them from both generalizing and disambiguating homophones/homonyms. This would be instead possible using an ST system, but requires a reliable recognition and translation of NEs and terms, without generating wrong and potentially-harmful suggestions (Stewart et al., 2018).

In light of the above, in this chapter we first assess the ability of direct and cascade ST systems in translating NEs and terminology (§6.3). Then, we focus on increasing the performance of the systems for the category that turned out as the most difficult to handle for ST systems: person names. On one side, we study the factors that influence the ability of ST systems in translating person names and how to act accordingly toward improving accuracy (§6.4). On the other, we also explore how to leverage external, contextual knowledge in the form of dictionaries of NEs that pertain to a specific domain, which are often curated by professional translators and interpreters (§6.5). At last, we investigate models that are capable of jointly translating from speech and recognizing NEs, comparing them with a pipeline of ST and NE recognition (NER) tools, in terms of both effectiveness in addressing the tasks and computational efficiency (§6.6).

As such, the contributions of this chapter are: *i)* the introduction of the first benchmark to assess NE and terminology translation quality for ST; *ii)* the demonstration that the nationality of the referent and the frequency in the training set are critical factors for the correct processing of person names, and that multilingual models predicting both transcripts and translations (attending also to the transcript) improve person name accuracy; *iii)* the first solution for direct ST that identifies which entities in a dictionary are present in an utterance and conditions the output generation on their translation; *iv)* the introduction of the first models able to jointly translate and recognize NEs, without introducing significant computational overhead with respect to a plain direct ST model.

6.2 Related Works

To the best of our knowledge, the work presented in this chapter is the first that addresses the topic of NE and terminology translation from speech. For this reason, this section overviews relevant papers that assess and try to improve NE and terminology translation in the related field of NMT (§6.2.1), investigate the recognition of person names and other NEs in ASR (§6.2.2), and integrate (or inject) external knowledge in ASR models (§6.2.3).

6.2.1 NE and Terminology in NMT

The translation of rare words – as NEs and specific terms are – is one of the main challenges for NMT models (Sennrich et al., 2016; Koehn and Knowles, 2017). In the case of NEs, researchers have hence confronted with the topic for many years, opening different lines of research: *i)* the substitution of NE with placeholders; *ii)* the addition of information about NE categories to the source sentence; *iii)* the inclusion of auxiliary loss functions dedicated to NER while training NMT models; *iv)* the integration of knowledge graphs (KGs).

The substitution of NE with placeholders (*i)*) has been explored by Post et al. (2019) that replace the NEs identified using regular expressions (e.g., for email or URL) or a dictionary before feeding the source sentence to the NMT model. The model is trained to produce special placeholders indicating the type of entity and the position in the target sentence, which are then replaced with the corresponding entry in the dictionary or the source representation in the case it was identified with a regular expression.

The addition of information about NE categories to the source sentence (*ii)*) has been instead tested by Ugawa et al. (2018); Zhou et al. (2020) that enrich the source sentence with start/end NE tags, and by Modrzejewski et al. (2020) that sum the token embeddings of the source sentence with embeddings of the NE type they belong to. As all these solutions require the annotation of the source sentence with a NER model, they introduce a significant computational overhead.

To avoid this issue, (Xie et al., 2022) recently proposed the inclusion of auxiliary loss functions dedicated to NER while training NMT models (*iii)*), as they add losses for the NER task both on the encoder output (recognizing the NEs in the source) and on the decoder output (recognizing the NEs in the target). Their experiments claim the superiority of this approach over the addition of information about NE categories to the source sentence (*ii)*), not only in terms of inference time, but also from the qualitative standpoint in the English-Chinese and Japanese-English language pairs.

Lastly, in the integration of KGs (*iv)*), the main goal is to transfer the knowledge present in KG into NMT models. Moussallem et al. (2019) concatenate embeddings of KG entities to the encoder and decoder embeddings of NMT systems, while Lu et al. (2019b); Zhao et al. (2020a,b); Ahmadnia et al. (2020) exploit KGs to enforce that the embeddings of the entities are similar in the source and target side, forcing the relationship between linked entities, and introduce synthetic training data built to contain parallel entities of the KG.

Similarly, most approaches devised to tackle terminology translation exploit domain-specific bilingual dictionaries (Hokamp and Liu, 2017; Chatterjee et al., 2017; Hasler et al., 2018; Dinu et al., 2019; Song et al., 2020; Dougal and Lonsdale, 2020; Niehues, 2021). In this case, the source sentence is enriched with the translation of each term found in a dictionary (e.g., "this is a term"→"this is a #term#*término*#"), before being fed to the NMT model.

Unfortunately, all the above-mentioned techniques assume that the source sentence is represented as text. Indeed, they are all based on NER tools that recognize which words belong to NEs, and/or on textual matching with other sources (e.g., dictionaries, KGs). Since there is no existing method to identify which portions of an audio segment correspond to NEs and which do not, none of these solutions is applicable to our direct ST scenario. As such, to the best of our knowledge, in §6.3 we investigate for the first time the ability of ST systems in handling NEs, and in §6.4 and §6.5 we describe the first attempts to improve the performance of direct ST systems on handling NEs, which cannot be compared these solutions.

6.2.2 NER from speech

As we just mentioned, handling NEs is even more challenging when the source modality is audio, as in the case of ST and ASR. To foster research on this problem, Galibert et al. (2014) introduced the first benchmark for NER recognition from speech, extracted from recordings of French TV and radio shows. A similar dataset has been released by Yadav et al. (2020), who manually annotated English ASR corpora from different domains (e.g., TED talks, audiobook readings) with NEs. However, none of these benchmarks is suitable for ST, as they are both limited to ASR (i.e., they only include the transcript).

Despite the presence of these suitable ASR benchmarks with NE annotations, few works approached the topic. Ghannay et al. (2018), Caubrière et al. (2020), Porjazovski et al. (2021), and Chen et al. (2022) mostly compare pipelines of ASR and NER tools with end-to-end models that directly extract NEs (in some cases the output is the transcript with inline NE tags, in others – e.g., Ghannay et al. 2018 – characters outside NEs are ignored). The conclusions of these works are contradicting with respect to which approach is best (end-to-end or cascade), but the paradigms are always close and, overall, can be considered on par in terms of performance. To the best of our knowledge, none of the existing works assesses the ability in translating NEs, as we do in §6.3, and the recognition of NEs on textual translations of audio content has been neglected before our analysis described in §6.6.

6.2.3 Knowledge Integration in ASR

The topic of contextual knowledge injection into ASR systems has mostly targeted the application of voice-command-recognition task, where user-specific content, such as contact names and application names, has to be correctly processed (Raghavan and Allan, 2005; Suchato et al., 2011; Bruguier et al., 2016). Without the sake of completeness, this section summarizes the most relevant research directions that influenced our work in §6.5.

One line of research concentrated on improving the recognition of user-specific content by creating dedicated language models. Such direction has been investigated for the first time in the context of traditional ASR hybrid systems, where the acoustic modeling is performed with neural Hidden Markov Models (HMM) and the language is modeled with a finite state transducer (FST). The works in this area (Novak et al., 2012; Aleksic et al., 2015; Williams et al., 2018; Ravi et al., 2020; Jung et al., 2022) build adapted FST that assign a high probability to user-specific content and mostly differ in the way such FSTs are built or in the way they are used to rescore the hypotheses. With the same principle, in end-to-end ASR Toshniwal et al. (2018) proposed the shallow fusion integration that rescores all the tokens in the hypotheses of the ASR model with an adapted LM.

These solutions have been further improved by proposing a class-based rescoring (Chen et al., 2019; Zhao et al., 2019; Huang et al., 2020; Gourav et al., 2021; Sun et al., 2021). In this case, a different LM is built for each class (or category of entities, such as person names and application names), and a rule or condition for its activation is defined. At inference time, when an activation condition is met in a hypothesis, its tokens are rescored with the corresponding class language model. The works in this direction mostly differ in the way the activation conditions are defined. In §6.5, when adopting the class-based LM rescoring in ST as a baseline, we follow Huang et al. (2020), who relabeled the speech training data to insert (start and end) class tags into the target text and activated the corresponding class LM at inference time when a candidate contains its start class tag.

A different approach has been taken by Pundak et al. (2018), who proposed CLAS (Contextual Listen Attend and Spell) to integrate domain-specific information into end-to-end ASR models. Specifically, the authors add a context (or bias) encoder, which first builds an embedding for each context sentence and then concatenates them together with a *no-bias* vector, and the decoder attends to the outputs of both the acoustic encoder and the bias encoder. To avoid the degradation that occurs in the presence of many contextual sentences, they also created a rule-based system that generates prefixes for each sentence, and assigns 0 probability to all the sentences whose prefix is not present in the hypothesis. The success of the CLAS method has led to its adaptation to different architectures, such as transducer models Chang et al. (2021); Jain et al. (2020); Chang et al. (2021). In our attempt – the first, to the best of our knowledge – of integrating external contextual knowledge in direct ST models (§6.5), we adapt the CLAS architecture to the Transformer decoder with the goal of injecting the translation of the NEs considered present in an utterance. However, the language switch between the source and target representations in ST first requires the identification of the entities present in the source utterance, which is our focus in §6.5, as the solutions proposed in MT (see §6.2.1) are not feasible due to the different input modality.

6.3 Named Entities and Terminology in ST

The assessment of the capability of state-of-the-art ST systems to properly translate NEs and terminology present in an utterance is hindered by the dearth of publicly available resources tailored to their specific evaluation. Therefore, our investigation on the topic started with the creation of a dedicated benchmark. Building on the newly-created resource, since the long dominance of cascade systems has gradually diminished thanks to the huge improvements of the direct approach, as described in §3, we then focused on understanding whether the inherent strengths and weaknesses of the two paradigms (Sperber and Paulik, 2020) can favor one or the other when it comes to the translation of NEs and terms. Indeed, while direct ST models avoid error propagation and can take advantage of unmediated access to audio information (e.g., prosody) during the translation phase, cascade solutions can exploit sizeable datasets for the ASR and MT subcomponents.

All in all, the contributions of this line of work are: *i)* the release NEuRoparl-ST, a novel benchmark on three language directions (en→es/fr/it) built from European Parliament speeches annotated with NEs and terminology, and *ii)* the first systematic analysis of the behavior of state-of-the-art ST systems in translating NEs and terminology. Our experiments show that direct and cascade ST systems display very similar behaviors, although direct models have slightly higher scores on NEs and slightly lower on terms. Overall, both cascade and direct ST systems prove to struggle more with NEs, as they correctly translate 75–80% of terms and 65–70% of NEs, with very low performance (37–40%) on person names.

6.3.1 Evaluation Data: NEuRoparl-ST

To the best of our knowledge, freely available NE/term-labelled ST benchmarks suitable for our analysis did not exist at the beginning of this PhD. The required resource should contain *i)* the audio corresponding to an utterance, *ii)* its transcript, *iii)* its translation in multiple target languages (three in our case), and *iv)* NE/term annotation in both transcripts and target texts. Currently available MT, ST, ASR, NE and terminology datasets lack at least one of these key components. For example, most MT corpora (e.g., Europarl) lack both audio sources and NE/terminology annotations. The very few available MT corpora annotated with NE/terminology still lack the audio portion, and extending them to ST would require generating synthetic audio, which is known to be problematic for the performance of ST models. For these reasons, we preferred to create a benchmark by annotating the en→es/fr/it transcripts and translations of the Europarl-ST test sets, which are mainly derived from the same original speeches. The result, NEuRoparl-ST,[2] is a multilingual benchmark featuring very high content overlap, thus enabling cross-lingual comparisons.

[2] Available at `https://ict.fbk.eu/neuroparl-st/`.

NE annotation. To build NEuRoparl-ST, we used the 18 tags and the annotation scheme defined by the guidelines ("OntoNotes Named Entity Guidelines - Version 14.0") used to annotate the OntoNotes5 corpus (Weischedel et al., 2012). The annotation was carried out manually by a professional interpreter with a multi-year experience in translating from English, French, and Italian into Spanish the verbatim reports of the European Parliament plenary meetings. This guarantees the high level of language knowledge and domain expertise required to achieve maximum quality and precision. To ease the task, the annotator was provided with transcripts and translations automatically pre-annotated with the BERT-based NER model[3] available in DeepPavlov (Burtsev et al., 2018). Human annotation was then conducted in parallel on the three test sets by labeling, for each audio segment, the English transcript and the three corresponding translations. To check the reliability of the annotations, all English transcripts were also independently labeled by a second annotator with a background in linguistics and excellent English knowledge. The inter-annotator agreement was calculated in terms of *complete* agreement, i.e. the exact match of the whole NE in the two annotations. The resulting Dice coefficient[4] (Dice, 1945) amounts to 93.87%, which can be considered highly satisfactory. For the subset of NEs for which complete agreement was found (1,409 in total), we also computed the agreement on label assignment with the *kappa coefficient* (in Scott's π formulation – Scott 1955). The resulting value is 0.94, which corresponds to "almost perfect" agreement according to its standard interpretation (Landis and Koch, 1977).

[3] `http://docs.deeppavlov.ai/en/master/features/models/ner.html`

[4] Note that Dice coefficient has the same value of the F1 measure computed considering either annotator as reference.

	en-es		**en-fr**		**en-it**	
	en	**es**	**en**	**fr**	**en**	**it**
NEs	1,637 (2,703)	1,638 (3,003)	1,578 (2,604)	1,562 (2,949)	1,523 (2,497)	1,466 (2,649)
TERMS	2,571 (3,174)	2,662 (3,294)	2,797 (3,502)	2,947 (3,659)	2,166 (2,669)	2,202 (2,645)
Num. of sentences	1,267		1,214		1,130	

Table 6.1: Total number of named entities and terms annotated in the test sets (and the corresponding number of tokens).

Terminology annotation. Similar to (Dinu et al., 2019), terminology was automatically extracted by exploiting the IATE termbase.[5] Each entry in IATE has an identifier and a language code. Entries with the same identifier and different language codes represent the translations of a term in the corresponding languages. To annotate our parallel texts, we first removed stop-words and lemmatized the remaining words and IATE entries.[6] Then, for each parallel sentence, we marked as terms only those words in the source and the target side that were present in IATE with the same identifier. This source/target match is essential to avoid the annotation of words that are used with a generic, common meaning but, being polysemic, can be technical terms in different contexts (e.g., the word "board" can refer to a tool or to a committee). Checking the presence of the corresponding translation in the target language disambiguates these cases, leading to a more accurate annotation.

NE and term annotations were merged into a single test set using BIO (Ramshaw and Marcus, 1995) as span labeling format. Had a word been tagged both as term and NE, the latter was chosen, favoring the more reliable manual annotation. Table 6.1 presents the total number of NEs and terms for the three language pairs, together with their corresponding number of tokens. These numbers differ between source and target texts and across pairs due to the peculiarities of the Europarl-ST data. Specifically, *i)*

[5] `http://iate.europa.eu`

[6] Preprocessing made with spaCy: `http://spacy.io/`

	en-es		**en-fr**		**en-it**	
	en	**es**	**en**	**fr**	**en**	**it**
CARDINAL	91 (105)	85 (104)	87 (101)	90 (105)	86 (100)	85 (98)
DATE	149 (314)	152 (321)	145 (303)	144 (377)	141 (300)	141 (294)
EVENT	8 (26)	9 (27)	7 (22)	7 (27)	8 (26)	9 (31)
FAC	18 (31)	19 (38)	18 (31)	21 (52)	18 (31)	16 (33)
GPE	241 (338)	240 (361)	232 (322)	221 (312)	222 (316)	209 (300)
LANGUAGE	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)
LAW	146 (478)	141 (622)	136 (448)	143 (608)	137 (439)	128 (509)
LOC	96 (121)	91 (122)	92 (118)	86 (128)	89 (111)	83 (109)
MONEY	10 (34)	11 (45)	10 (34)	11 (49)	6 (20)	6 (25)
NORP	135 (151)	126 (147)	136 (151)	156 (182)	123 (139)	143 (194)
ORDINAL	64 (64)	65 (65)	57 (57)	40 (40)	62 (62)	52 (53)
ORG	565 (857)	582 (989)	550 (844)	533 (906)	520 (773)	485 (851)
PERCENT	4 (10)	4 (6)	3 (8)	3 (9)	4 (10)	4 (14)
PERSON	92 (134)	96 (122)	88 (129)	88 (122)	89 (130)	87 (101)
PRODUCT	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)
QUANTITY	3 (7)	3 (7)	3 (7)	3 (7)	3 (7)	3 (7)
TIME	11 (26)	10 (20)	10 (22)	9 (18)	11 (26)	11 (24)
WORK_OF_ART	1 (4)	1 (4)	1 (4)	1 (4)	1 (4)	1 (3)
TERM	2571 (3174)	2662 (3294)	2797 (3502)	2947 (3659)	2166 (2669)	2202 (2645)

Table 6.2: Number of named entities and terms annotated in the test sets (and the corresponding number of tokens).

sometimes translations are not literal and NEs are omitted in the translation (e.g., when a NE is repeated in the source, one of the occurrences may be replaced by a pronoun in the target text), *ii)* the professional interpreters and translators "localize" the target translations, i.e. adapt them to the target culture (e.g., while the English source simply contains the name and surname of mentioned European Parliament members, in Italian the first name is omitted and the surname is preceded by "onoverole" - honorable), and *iii)* the number of words a NE is made of can vary across languages (e.g "European Timeshare Owners Organisation" becomes "Organización Europea de Socios de Tiempo Compartido" in Spanish).

Table 6.2 presents the number of named entities (NEs) and terms annotated in the test sets, divided by category. Since both NEs and terms can be composed of more than one word (e.g., for a person it is common to have both the name and surname), the total number of tokens per category is also given.

Enabled by this newly created benchmark, we proceed with a systematic analysis of the behavior of cascade and direct ST systems to answer understand whether their behavior is similar on not with respect to NE and terminology translation.

6.3.2 Experimental Settings

To compare the cascade and direct solutions, we build strong models trained on large corpora, which are described below, after the details regarding the evaluation procedure.

Data and Evaluation

As ASR training data, we used LibriSpeech, TEDLIUM v3, and Mozilla Common Voice, together with (*utterance*, *transcript*) pairs extracted from three ST corpora: MuST-C, Europarl-ST, and CoVoST 2. We augment data with SpecAugment and, after lowercasing and punctuation removal, the text is split into sub-words with 8,000 BPE merge rules. The same datasets are used for training our direct ST models, where the transcripts contained in the ASR training corpora synthetically are translated with our NMT model.

The MT training data were collected from the OPUS repository and cleaned with the ModernMT framework. At the end of this process, the actual training data is reduced to 45M segment pairs (550M English words) for English-Italian. For English-Spanish, the training data is further filtered with data selection methods (Axelrod et al., 2011) using a general-domain seed resulting in 19M segment pairs (330M English words). Finally, for English-French we have 28M sentence pairs (550M of English words).

We use our benchmark to measure the ability of systems in handling NEs and terminology. Transcription and translation quality are respectively measured with WER and SacreBLEU[7]. Similarly to the Named Entity Weak Accuracy proposed in (Hermjakob et al., 2008), we compute NE/term accuracy[8] as the ratio of entities that are present in the output of the evaluated system in the correct form.

Architectures and Training Details

Cascade ST Model The **ASR** component of our cascade is a Transformer-based model consisting of 11 encoder layers, 4 decoder layers, 8 attention heads, 512 features for the attention layers and 2,048 hidden units in the feed-forward layers. Its encoder has been adapted for processing speech by means of two initial 2D convolutional layers that reduce the input sequence length by a factor of 4. Also, the encoder self-attentions are biased using a logarithmic distance penalty that favors the local context. The model is trained with an additional Connectionist Temporal Classification (CTC) loss, which is added as a linear layer to the 8th encoder layer. We set the dropout to 0.1. We optimize label-smoothed cross entropy with a smoothing factor of 0.1 with Adam. The learning rate is increased for 5,000 steps from 0.0003 up to 0.0005 and then decays with inverse square root policy. Our mini-batches are composed of up to 12K tokens or 8 samples and we delay parameter updates for 8 mini-batches, training on 8 K80 GPUs.

Before feeding the MT with the ASR outputs, the transcripts are post-processed by an additional model to restore casing and punctuation. This model is a Transformer-based system trained on data from the OPUS repository, where the source text is lowercase and without punctuation, and the target text is made of normally formatted sentences.

[7] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0`

[8] Scores have been computed with the script available at `https://github.com/mgaido91/FBK-fairseq-ST/blob/emnlp2021/scripts/eval/ne_terms_accuracy.py`

	en-es				**en-fr**				**en-it**			
	WER	BLEU	NE	Term	WER	BLEU	NE	Term	WER	BLEU	NE	Term
ASR	12.6	–	84.6	92.6	12.7	–	84.5	92.1	12.6	–	84.3	92.4
MT	–	48.8	83.5	88.8	–	36.2	78.8	85.7	–	33.8	80.2	86.9
Cascade	–	37.6	70.9	82.5	–	28.3	66.7	80.4	–	26.5	66.9	80.4
Direct	–	37.7	71.4	79.2	–	30.1	67.3	77.7	–	26.0	67.3	76.3

Table 6.3: WER/BLEU and NE/term case-insensitive accuracy for ASR, MT and ST (cascade and direct) models.

The **MT** component is a Transformer model with 6 layers for both the encoder and the decoder, 16 attention heads, 1,024 features for the attention layers, and 4,096 hidden units in the feed-forward layers. Models are optimized on label-smoothed cross entropy with Adam, with a learning rate that linearly increases for 8,000 updates up to 0.0005, after which decays with inverse square root policy. Each batch is composed of 4 mini-batches made of 3072 tokens. Dropout is set to 0.3. We train for 200,000 updates and average the last 10 checkpoints. Source and target languages share a BPE vocabulary of 32k sub-words.

Direct ST Model Our **direct** model has the same architecture as the ASR component described above, which is also used to initialize its encoder weights. Besides encoder pre-training, for knowledge transfer, we also distill knowledge from the MT model with the three-step process introduced in §3.3. In addition, we use SpecAugment and time stretch.

6.3.3 Results

In this section, we compare the two ST systems just described, and we also analyze the ASR and MT subcomponents (the latter being fed with human transcripts) of the cascade system. Table 6.3 presents the case-insensitive scores to fairly compare the different models, as the ASR produces lowercase text. For the sake of completeness, case-sensitive NE/term accuracy is also

	en-es		**en-fr**		**en-it**	
	NE	Term	NE	Term	NE	Term
MT	81.0	88.0	75.5	85.3	77.5	86.2
Cascade	65.8	81.6	61.3	79.9	62.6	79.5
Direct	69.4	78.7	65.9	77.3	65.1	75.9

Table 6.4: Case-sensitive accuracy scores of MT and ST (cascade and direct) models on en→es/fr/it.

given in Table 6.4 for ST and MT models (we do not include ASR since it generates lowercase text). Comparing these results with those reported in Table 6.3, for all language pairs we see that the drop in NEs accuracy with respect to case-insensitive scores is higher for the cascade model – around 5 points – than for the direct one – around 2 points (e.g., for en-es, from 70.9 to 65.8 for the cascade model and from 71.4 to 69.4 for the direct model). We posit the reason is the propagation of errors in the module in charge to restore casing on the ASR output in the cascade architecture.

ASR and MT results

The WER of the ASR is similar across the three language directions. This is not surprising because the three test sets differ only in very few debates. In terms of accuracy, it is evident that transcribing NEs is more difficult than transcribing terms (average accuracy: 84.5 vs 92.4). Besides lower frequency, the higher difficulty to transcribe NEs can be ascribed to the variety of different pronunciations by non-native speakers (in particular for person, product, and organization names). Concerning the MT performance, the BLEU differences between language directions (en-es ≫ en-fr > en-it) reflect the results reported in the Europarl-ST paper (Iranzo-Sánchez et al., 2020). The main reason is that the translations are less literal for some language directions. For instance, the French references are 20% longer than the human source transcripts. Analyzing NE and term translation quality, we notice that NEs are, again, harder to handle compared to terminology (average accuracy: 80.8 vs 87.1). It is worth noticing that accuracy does not strictly depend on translation quality. For instance, en-fr has a higher translation quality than en-it (+2.4 BLEU points), but NE and term accuracy scores are lower.

ST results

Unsurprisingly, when it comes to combining transcription and translation in a single task, performance decreases significantly. In particular, the results of the cascade model are a direct consequence of cumulative ASR and MT errors. As such, like for its sub-components, NEs are harder to handle than terms. Compared to the MT results computed on manual transcripts, we see large drops on all languages in both translation quality (-13.2 BLEU on average) and NE/term accuracy (-12.8/-6.0).

Comparing cascade and direct models, the BLEU scores are on par for en-es and en-it (differences are not statistically significant[9]), while the direct one is significantly better for en-fr. This is explained by the aforementioned peculiarity of the French reference translations in Europarl-ST that, unlike in common training corpora (Europarl included), are on average 20% longer than the source transcripts. The MT model of the cascade, trained on massive corpora including Europarl, tends to produce translations that are similar in length to the transcripts and shorter than Europarl-ST references, being thus penalized. Having Europarl-ST among its training corpora, the direct model produces outputs more similar in length to the references, resulting in a 2.8 BLEU gain.

In terms of NE and term translation quality, the trend is clear and coherent in all languages: the cascade outperforms the direct on terminology (+3.5 on average), while the direct has an edge (+0.5) in handling NEs. The advantage of the cascade on terminology can be explained by the higher

[9] Computed with bootstrap resampling (Koehn et al., 2003).

	en-es				**en-fr**				**en-it**			
	ASR	**MT**	**Casc.**	**Dir.**	**ASR**	**MT**	**Casc.**	**Dir.**	**ASR**	**MT**	**Casc.**	**Dir.**
CARDINAL	92.31	88.24	80.00	76.47	94.25	86.67	80.00	81.11	93.02	89.41	78.82	80.00
DATE	90.60	78.95	73.68	72.37	89.66	57.64	52.08	56.25	89.36	76.60	67.38	68.09
EVENT	37.50	33.33	33.33	33.33	28.57	71.43	28.57	57.14	37.50	66.67	44.44	55.56
FAC	77.78	73.68	63.16	57.89	77.78	57.14	52.38	47.62	77.78	75.00	62.50	50.00
GPE	94.61	84.17	79.17	82.50	94.40	89.19	81.98	86.88	94.62	89.00	83.25	82.30
LANGUAGE	100.00	100.00	100.00	100.00	100.00	100.00	100.00	50.00	100.00	100.00	100.00	50.00
LAW	69.86	63.12	46.10	42.55	70.59	65.73	46.15	43.36	69.34	69.53	53.13	47.66
LOCATION	93.75	85.71	79.12	81.32	92.39	81.40	77.91	74.42	93.26	79.52	79.52	74.70
MONEY	20.00	54.55	27.27	72.73	20.00	18.18	27.27	27.27	16.67	66.67	16.67	66.67
NORP	87.41	79.37	70.63	69.84	87.50	69.43	63.06	62.18	86.99	70.63	60.84	56.64
ORDINAL	90.63	81.54	72.31	69.23	89.47	80.00	65.00	70.00	90.32	80.77	65.38	65.38
ORG	89.38	89.00	77.49	78.69	89.09	84.08	73.97	73.36	89.23	79.59	67.63	71.96
PERCENT	0.00	100.00	0.00	75.00	0.00	66.67	0.00	66.67	0.00	25.00	0.00	75.00
PERSON	40.22	93.75	40.63	38.54	39.77	93.18	38.64	38.64	39.33	98.85	42.53	41.38
PRODUCT	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
QUANTITY	0.00	66.67	0.00	0.00	0.00	100.00	33.33	33.33	0.00	66.67	0.00	33.33
TIME	63.64	100.00	80.00	70.00	60.00	77.78	77.78	66.67	63.64	63.64	63.64	45.45
WORK_OF_ART	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 6.5: Case insensitive accuracy scores for all the NE types on the three language pairs. We report the results for ASR, MT, Cascade (Casc.) and Direct (Dir.) systems.

reliability of its MT component in selecting domain-specific target words compared to the direct models built on much smaller ST training corpora. One example is the English term “plastic explosive”, which is correctly translated into Italian by the cascade (“esplosivo plastico”), and wrongly by the direct (“esplosivo di plastica” - En: “explosive made of plastic”). Concerning NEs, instead, the unmediated access to the audio helps the direct to avoid both *i)* error propagation (e.g., the NE “Lamfalussy” is correctly translated by the direct, while the MT component of the cascade is not able to recover the wrong ASR output “blunt Hallucy”), and *ii)* the translation of NEs that are homographs of common nouns in the source language but should be copied *as is* (e.g., the English surname “Parish” is translated into Italian as “Parrocchia” by the cascade, but correctly preserved in the output of the direct model).

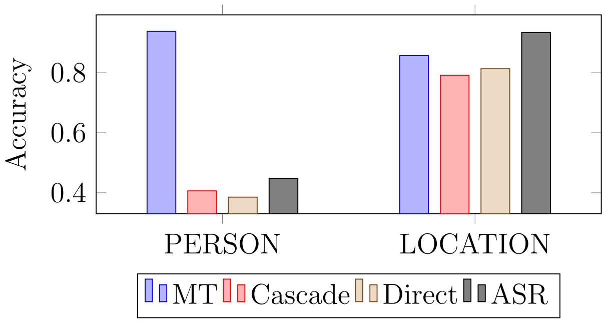


Figure 6.1: Accuracy scores on PERSON and LOCATION of MT, ASR and ST systems on en-es.

Looking at NE types (Table 6.5), we can notice that the performance of the two ST systems (cascade and direct) are similar in all categories, except for some types (e.g., LANGUAGE, PRODUCT, QUANTITY) that show a high variability caused by the limited number of examples with that label. This demonstrates that their different architecture does not bring a different ability in handling a specific type of entity, reflecting the global accuracy scores in Table 6.3. For both approaches, the differences across the NE types depend on their capability to *recognize* entities in the audio and properly *translate* them. Two types are paradigmatic (see Figure 6.1). PERSON names (the worst category, with 37–40% ST accuracy) are difficult to recognize in the audio, as shown by the poor performance of ASR and both ST systems, while their translation from manual transcripts (MT) is trivial as it only requires copying them from the source. Conversely, ST and MT results are very close on the more frequent and normally easier-to-pronounce LOCATION names, for which the problem lies more in translation than in recognition.

6.3.4 Summary

We started our investigation toward augmented ST by addressing the foremost limitation for the understanding of the behavior of ST systems with respect to NEs and terminology: the dearth of suitable labeled benchmarks. To fill this gap, we created an annotated test set, NEuRoparl-ST, covering three language directions, and used it for the first comparison of state-of-the-art cascade and direct ST systems on NE and term translation. Our results show that NEs, and especially person names, are in general more difficult to handle than terminology. For this reason, in the next sections we analyze which factors hinder the correct handling of person names and subsequently propose solutions to improve person-name accuracy. In particular, we address two scenarios: when a dictionary of names likely to occur in a domain is present (§6.5), and when it is not (§6.4).

6.4 Handling Person Names in ST

Following the finding of the previous section regarding the poor handling of person names by ASR/ST systems, we move on to: *i)* identify the factors causing the problem, and *ii)* design appropriate mitigation solutions. Toward these objectives, our first contribution (§6.4.1) is the annotation[10] of each person name occurring in NEuRoparl-ST with their nationality and the nationality of the speaker (as a proxy of the native language) – e.g., if a German person says "*Macron is the French president*", the speaker's nationality is German, while the referent nationality is French. Drawing on this additional information, our second contribution (§6.4.1) is the analysis of the concurring factors involved in the correct handling of person names. Besides the frequency, we identify as a key discriminating factor the presence in the training data of speech uttered in the referent's native language (e.g., French in the above example). This finding, together with an observed accuracy gap between person name transcription (ASR) and translation (ST), leads to our third contribution (§6.4.3): a multilingual ST system that

[10] Available at: `https://ict.fbk.eu/neuroparl-st/`.

jointly transcribes and translates the input audio, giving higher importance to the transcription task in favor of a more accurate translation of person names. Our model shows relative gains in person name translation accuracy by 48% on average on three language pairs (en→es,fr,it), producing useful translations for interpreters in 66% of the cases.

6.4.1 Factors Influencing Name Recognition

As shown in the previous section, the translation of person names is difficult both for direct and cascade ST systems, which obtain similar accuracy scores (∼40%). The low performance of cascade solutions is largely due to errors made by the ASR component, while the MT model usually achieves nearly perfect scores. For this reason, henceforth we will focus on identifying the main issues related to the transcription and translation of person names, respectively in ASR and direct ST.

We hypothesize that three main factors influence the ability of a system to transcribe/translate a person name: *i)* its frequency in the training data, as neural models are known to poorly handle rare words, *ii)* the nationality of the referent, as different languages may involve different phoneme-to-grapheme mappings and may contain different sounds, and *iii)* the nationality of the speaker, as non-native speakers typically have different accents and hence different pronunciations of the same name. To validate these hypotheses, we inspect the outputs of Transformer-based ASR and ST models trained with the configuration defined in (Wang et al., 2020b). For the sake of reproducibility, complete details on our experimental settings are provided in §6.4.2.

Data and Annotation

To enable fine-grained evaluations on the three factors we suppose to be influential, we enrich the NEuRoparl-ST benchmark by adding three (one for each factor) features to each token annotated as *PERSON*. These are: *i)* the token frequency in the target transcripts/translations of the training set, *ii)* the nationality of the referent, and *iii)* the nationality of the speaker. The nationality of the referents was manually collected through online searches. The nationality of the speakers, instead, was automatically extracted from the personal data listed in LinkedEP (Hollink et al., 2017) using the country they represent in the European Parliament.[11] All our systems are trained on Europarl-ST and MuST-C, and evaluated on this new extended version of NEuRoparl-ST.

The Role of Frequency

As the first step in our analysis, we automatically check how the three features added to each *PERSON* token correlate with the correct generation of the token itself. To this end, we train a classification decision tree (Breiman et al., 1984). Classification trees recursively divide the dataset into two groups, choosing a feature and a threshold that minimize the entropy of the resulting groups with respect to the target label. As such, they do not assume a linear relationship between the input and the target (like multiple regression and random linear mixed effects do) and are a good fit for categorical features as most of ours are. Their structure makes them easy to interpret (Wu et al., 2008): the first decision (the root of the tree) is the most important criterion according to the learned model, while less discriminative features are pushed to the bottom.

We feed the classifier with 49 features, corresponding to: *i)* the frequency of the token in the training data, *ii)* the one-hot encoding of the speaker nationality, and *iii)* the one-hot encoding of the referent nationality.[12] We then train it to predict whether our ASR model is able to correctly

[11] For each speech in Europarl-ST, the speaker is referenced by a link to LinkedEP.

[12] Speakers and referents respectively belong to 17 and 31 different nations.

	All	**Freq. ≥ 3**	**Freq. < 3**
ASR	38.46	55.81	4.55
en-fr	28.69	45.45	0.00
en-es	35.29	53.57	19.05
en-it	29.70	46.77	2.56
Average	33.04	50.40	6.54

Table 6.6: Token-level accuracy of person names divided into two groups according to their frequency in the training set for ASR and ST (en→es/fr/it) systems.

transcribe the token in the output. To this end, we use the implementation of scikit-learn (Pedregosa et al., 2011), setting to 3 the maximum depth of the tree, and using Gini index as an entropy measure.

Unsurprisingly, the root node decision is based on the frequency of the token in the training data, with 2.5 as a split value. This means that person names occurring at least 3 times in the training data are likely to be correctly handled by the models. Although this threshold may vary across datasets of different sizes, it is an indication of the necessary number of occurrences of a person name, potentially useful for data augmentation techniques aimed at exposing the system to relevant instances at training time (e.g., names of famous people in the specific domain of a talk to be translated/interpreted). We validate that this finding also holds for ST systems by reporting in Table 6.6 the accuracy of person tokens for ASR and the three ST language directions, split according to the mentioned threshold of frequency in the training set. On average, names occurring at least 3 times in the training set are correctly generated in slightly more than 50% of the cases, a much larger value compared to those with fewer than 3 occurrences.

The other nodes of the classification tree contain less interpretable criteria, which can be considered as spurious cues. For instance, at the second level of the tree, a splitting criterion is "*is the speaker from Denmark?*" because the only talk by a Danish speaker contains a mention to *Kolarska-Bobinska*

Referent	**ASR**	**en-fr**	**en-es**	**en-it**	**Freq.**
UK	52.38	59.09	63.16	41.18	46.21
non-UK	35.78	22.00	30.00	27.38	21.96
All	38.46	28.69	35.29	29.70	25.65

Table 6.7: Token-level accuracy of ASR and ST (en-fr, en-es, en-it) systems for UK/non-UK *referents*.

that systems were not able to correctly generate. We hence decided to perform further dedicated experiments to better understand the role of the other two factors: referent and speaker nationality.

The Role of Referent Nationality

Humans often struggle to understand names belonging to languages that are different from their native ones or from those they know. Moreover, upon manual inspection of the system outputs, we observed that some names were "Englishized" (e.g. *Youngsen* instead of *Jensen*). In light of this, we posit that a system trained to recognize English sounds and to learn English phoneme-to-grapheme mappings might be inadequate to handle non-English names.

We first validate this idea by computing the accuracy for names of people from the United Kingdom[13] ("UK" henceforth) and for names of people from the rest of the World ("non-UK"). Looking at Table 6.7, we notice that our assumption seems to hold for both ASR and ST. However, the scores correlate with the frequency (Freq.) of names in the training set[14] as, on average, UK referents have more than twice the occurrences (46.21)

[13] We are aware that our annotation is potentially subject to noise, due to the possible presence of UK citizens with non-anglophone names. A thorough study of the best strategies to maximize the accuracy of UK/non-UK label assignment is a task *per se*, out of the scope of our work. By now, as a manual inspection of the names revealed no such cases in our data, we believe that the few possible wrong assignments do not undermine our experiments, nor the reported findings.

[14] Notice that the ASR and the ST training sets mostly contain the same data, so frequencies are similar in the four cases.

	ASR	**en-fr**	**en-es**	**en-it**	**Avg.**
UK	42.86	25.76	33.33	29.41	32.84
non-UK	29.05	22.67	23.33	19.44	23.62
∆Accuracy	13.81	3.09	10.00	9.97	9.22

Table 6.8: Token-level accuracy of UK/non-UK *referents* averaged over three runs with balanced training sets.

of non-UK referents (21.96). The higher scores for UK referents may hence depend on this second factor.

To disentangle the two factors and isolate the impact of referents' nationality, we create a training set with balanced average frequency for UK and non-UK people by filtering out a subset of the instances containing UK names from the original training set.[11] To ensure that our results are not due to a particular filtering method, we randomly choose the instances to remove and run the experiments on three different filtered training sets. The results for the three ST language pairs and ASR (see Table 6.8) confirm the presence of a large accuracy gap between UK and non-UK names (9.22 on average), meaning that the accuracy on non-UK names (23.62) is on average $\sim$ 30% lower than the accuracy on UK names (32.84). As in this case we can rule out any bias in the results due to the frequency in the training set, we can state that the nationality of the referent is an important factor.

The Role of Speaker Nationality

Another factor likely to influence the correct understanding of person names from speech is the speaker's accent. To verify its impact, we follow a similar procedure to that of the previous subsection. First, we check whether the overall accuracy is higher for names uttered by UK speakers than for those uttered by non-UK speakers. Then, to ascertain whether the results depend on the proportion of UK/non-UK speakers, we randomly create

Speaker	ASR	en-fr	en-es	en-it	Freq.
UK	41.03	32.43	36.84	29.41	34.55
non-UK	37.36	27.06	34.57	29.85	21.76
All	38.46	28.69	35.29	29.70	25.65

Table 6.9: Token-level accuracy of ASR and ST systems for names uttered by UK/non-UK *speakers*.

Speaker	ASR	en-fr	en-es	en-it	Avg.
UK	29.91	29.73	28.95	23.53	28.03
non-UK	33.33	22.75	25.51	19.40	25.25
ΔAccuracy	-3.42	6.98	3.43	4.13	2.78

Table 6.10: Token-level accuracy of person names uttered by UK/non-UK *speakers* averaged over three runs with balanced training sets.

three training sets featuring a balanced average frequency of speakers from the two groups.

Table 6.9 shows the overall results split according to the two groups of speaker nationalities. In this case, the accuracy gap is minimal (the maximum gap is 5.37 for en-fr, while it is even negative for en-it), suggesting that the speaker accent has a marginal influence, if any, on how ASR and ST systems handle person names.

The experiments on balanced training sets (see Table 6.10) confirm the above results, with an average accuracy difference of 2.78 for ASR and the three ST language directions. In light of this, we can conclude that, differently from the other two factors, speakers' nationality has negligible effects on the ASR/ST performance on person names.

6.4.2 Experimental Settings

The previous section has uncovered that only two of the three considered factors actually have a tangible impact: the frequency in the training set, and the referent nationality. A low frequency can be tackled either by collecting more data or by generating synthetic instances (Alves et al., 2020; Zheng et al., 2021). Fine-tuning the model on additional material is usually a viable solution in the use case of assisting interpreters and translators since, during their preparation phase, they have access to various sources of domain-specific information (Gile, 2009; Díaz-Galaz et al., 2015). Focusing on the second factor, we experiment with the creation of multilingual models that are more robust to a wider range of phonetic features and hence to names of different nationalities. In addition, we investigate the adoption of the so-called "triangle" architecture (Anastasopoulos and Chiang, 2018) to close the gap between ASR and ST systems attested in §6.3 and confirmed by the preliminary results shown in Table 6.6.

Data and Evaluation

We train our systems on MuST-C and Europarl-ST. When using multilingual models, the ST training data (*→es/fr/it) consists of the en→es/fr/it sections of MuST-C and the {nl, de, en, es, fr, it, pl, pt, ro}→es/fr/it sections of Europarl-ST. Notice that, in this scenario, the English source audio constitutes more than 80% of the total training data, as MuST-C is considerably bigger than Europarl-ST and the English speeches in Europarl-ST are about 4 times those in the other languages.[15] For ASR, we use the audio-transcript pairs of the *-it training set defined above.

We extract 80 features from audio segments, while the target text is segmented into BPE subwords using 8,000 merge rules with SentencePiece.

Transcription and translation quality are measured respectively with WER and SacreBLEU on both MuST-C and Europarl-ST test sets.[16] The person name accuracy, instead, is computed on NEuRoparl-ST as seen in the previous sections.

[15] For instance, in *-fr the training set amounts to 671 hours of audio, 573 (i.e., 83%) having English audio.

[16] `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0`

Architectures and Training Details

All our ASR and ST models share the same architecture. Two 1D convolutional layers with a Gated Linear Unit non-linearity between them shrink the input sequence over the temporal dimension, having 2 as stride. Then, after adding sinusoidal positional embeddings, the sequence is encoded by 12 Transformer encoder layers, whose output is attended by 6 Transformer decoder layers. We use 512 as Transformer embedding size, 2048 as the intermediate dimension of the FFNs, and 8 heads. In the case of the triangle model, we keep the same settings and the configurations are the same for the two decoders. We filter out samples whose audio segment lasts more than 30 s, normalize them at utterance level, and apply SpecAugment.

Models are optimized with Adam to minimize the label-smoothed cross entropy. The learning rate increases up to 1e-3 for 10,000 warm-up updates, then decreases with an inverse square-root scheduler. We train on 4 K80 GPUs, using mini-batches containing 5,000 tokens, and accumulating the gradient for 16 mini-batches. We average 5 checkpoints around the best on the validation loss.

6.4.3 Results

Toward our goal of improving person name translation, in this section we report the results of two different interventions. First, we explore whether multilingual models increase the robustness of ASR and ST systems to non-UK referents. Second, we propose the triangle architecture to close the gap between ASR and direct ST systems in terms of person name accuracy.

Increasing Robustness to non-UK Referents

As illustrated in §6.4.1, one cause of failure of our ASR/ST models trained on English audio is the tendency to force every sound to an English-like word,

	Monolingual				**Multilingual**				
	ASR	**en-fr**	**en-es**	**en-it**	**ASR**	**en-fr**	**en-es**	**en-it**	
	WER (↓)	**BLEU (↑)**			**WER (↓)**	**BLEU (↑)**			
Europarl-ST	13.65	32.42	34.11	25.72	13.29	33.92	35.59	26.55	
MuST-C	11.17	32.81	27.18	22.81	11.86	33.34	27.72	23.02	
	Token-level Person Name Accuracy (↑)								**Avg. Δ**
Overall	38.46	28.69	35.29	29.70	46.15	38.52	44.54	36.63	+8.43
UK	52.38	59.09	63.16	41.18	66.67	59.09	63.16	52.94	+6.51
non-UK	35.78	22.00	30.00	27.38	42.20	34.00	41.00	33.33	+8.84

Table 6.11: Transcription/translation quality and token-level person name accuracy, both overall and divided into UK/non-UK referents. *Avg.* Δ indicates the difference between multilingual and monolingual systems averaged over the ASR and the three ST directions.

distorting person names from other languages. Consequently, we posit that a multilingual system, trained to recognize and translate speech in different languages, might be more robust and, in turn, achieve better performance on non-English names. We test this hypothesis by training multilingual ASR and ST models that are fed with audio in different languages, and respectively produce transcripts and translations (into French, Italian, or Spanish in our case).

We analyze the effect of including additional languages both in terms of general quality and in terms of person name transcription/translation accuracy. Looking at the first two rows of Table 6.11, we notice that the improvements in terms of generic translation quality (BLEU) are higher on the Europarl-ST than on the MuST-C test set – most likely because the additional data belongs to the Europarl domain – while in terms of speech recognition (WER) there is a small improvement for Europarl-ST and a small loss for MuST-C. Turning to person names (third line of the table), the gains of the multilingual models (+8.43 accuracy on average) are higher and consistent between ASR and the ST language pairs.

By dividing the person names into the two categories discussed in §6.4.1 – UK and non-UK referents – the results become less consistent across language pairs. On ST into French and Spanish, the accuracy of UK names remains constant, while there are significant gains (respectively +12 and +11) for non-UK names. These improvements seem to support the intuition that models trained on more languages learn a wider range of phoneme-to-grapheme mappings and so are able to better handle non-English names. However, the results for ASR and for ST into Italian seemingly contradict our hypothesis, as they show higher improvements for UK names (∼11-14) than for non-UK names (∼6-7).

We investigate this behavior by further dividing the non-UK group into two sub-categories: the names of referents whose native language is included in the training set ("in-train" henceforth), and those of referents whose native language is not included in the training set ("out-of-train"). For in-train non-UK names, the monolingual ASR accuracy is outperformed by the multilingual counterpart by 16.66 (33.33 vs 49.99), i.e. by a margin higher than that for UK names (14.29). For the out-of-train names, instead, the gap between the monolingual ASR accuracy (36.71) and the multilingual ASR accuracy (39.24) is marginal (2.5). Similarly, for ST into Italian the in-train group accuracy improves by 8.70 (from 34.78 to 43.48), while the out-of-train group accuracy has a smaller gain of 4.92 (from 24.59 to 29.51). These results indicate that adding a language to the training data helps the correct handling of person names belonging to that language, even when translating/transcribing from another language. Further evidence is exposed in §6.4.4, where we analyze the errors made by our systems and how their distribution changes between a monolingual and a multilingual model.

Closing the Gap Between ASR and ST

The previous results – in line with those of §6.3 – reveal a gap between ASR and ST systems, although their task is similar when it comes to person names. Indeed, both ASR and ST have to recognize the names from the speech, and produce them as-is in the output. Contextually, section 6.3 showed that neural MT models are good at "copying" from the source or, in other words, at estimating $p(Y|T)$ – where Y is the target sentence and T is the textual source sentence – when Y and T are the same string. Hence, we hypothesize that an ST model can close the performance gap with the ASR by conditioning the target prediction not only on the input audio, but also on the generated transcript. Formally, this means estimating $p(Y|X, T')$, where T' denotes a representation of the generated transcript, such as the embeddings used to predict them; and this estimation is what the triangle actually does.

The triangle model is composed of a single encoder, whose output is attended by two decoders that respectively generate the transcript (ASR decoder) and the translation (ST decoder). The ST decoder also attends to the output embeddings (i.e., the internal representation before the final linear layer mapping to the output vocabulary dimension and softmax) of the ASR decoder in all its layers. In particular, the outputs of the cross-attention on the encoder output and the cross-attention on the ASR decoder output are concatenated and fed to a linear layer. The model is optimized with a multi-loss objective function, defined as follows:

$$L(X) = -\sum_{x \in X} \Big(\lambda_{ASR} * \sum_{t \in T_x} log(p_\theta(t_i|x, t_{i-1,\dots,0})) + \lambda_{ST} * \sum_{y \in Y_x} log(p_\theta(y_i|x, T, y_{i-1,\dots,0})) \Big) \quad (6.1)$$

where T is the target transcript, Y is the target translation, and x is the input utterance. λ_{ASR} and λ_{ST} are two hyperparameters aimed at controlling the relative importance of the two tasks. Previous works commonly set them to 0.5, giving equal importance to the two tasks (Anastasopoulos and Chiang, 2018; Sperber et al., 2020). To the best of our knowledge, ours is the first attempt to inspect performance variations in the setting of these

Model	**WER (↓)**	**BLEU (↑)**			**Person Accuracy**					
	ASR	**en-es**	**en-fr**	**en-it**	**ASR**	**en-es**	**en-fr**	**en-it**	**ST Avg.**	**ASR-ST**
Base	13.29	35.86	33.99	26.80	46.15	44.54	38.52	36.63	39.90	6.25
Triangle	14.25	37.42	35.44	28.20	42.31	43.70	41.80	41.58	42.36	-0.05
λ_{ASR}=0.8, λ_{ST}=0.2	13.75	36.48	34.85	27.30	47.69	44.54	43.44	50.50	46.16	1.53

Table 6.12: WER (for ASR), SacreBLEU (for ST), and token-level person name accuracy computed on the NEuRoparl-ST test set. For triangle models, ASR scores are computed on the transcript output of the *-it model. *ST Avg.* is the average accuracy on the 3 language pairs (en→es,fr,it) and *ASR-ST* is the difference between the ASR and the average ST accuracy.

two parameters, calibrating them toward the specific needs arising from our application scenario.

In Table 6.12, we compare the previously-introduced multilingual models (*Base*) with triangle ST multilingual models trained on the same data (*Triangle*). Although the transcripts of the triangle models (second row) are less accurate (about +1 WER), the translations have higher quality (+1.4-1.6 BLEU on the three language pairs). Person names follow a similar trend: in the transcript, the accuracy is lower (-3.84), while in ST it increases (on average +2.46). Interestingly, the accuracy gap between ASR and ST is closed by the triangle model (see the ASR-ST column), confirming our assumption that neural models are good at copying. However, due to the lower ASR accuracy (42.31), the ST accuracy (42.36) does not reach that of the base ASR model (46.15). The reason for this drop can be found in the different types of information required by the ASR and ST tasks. Chuang et al. (2020) showed that the semantic content of the utterance is more important for ST, and that joint ASR/ST training leads the model to focus more on the semantic content of the utterance, yielding BLEU gains at the expense of higher WER. As person names are usually close in the semantic space (Das et al., 2017), the higher focus on semantic content may be detrimental to their correct handling and hence explain the lower person name accuracy.

In light of this observation, we experimented with changing the weights of the losses in the triangle training, assigning higher importance to the ASR loss (third row of Table 6.12). In this configuration, as expected, transcription quality increases (-0.5 WER) at the expense of translation quality, which decreases (-0.8 BLEU on average) but remains higher than that of the base model. The accuracy of person names follows the trend of transcription quality: the average accuracy on ST (46.16) increases by 3.8 points over the base triangle model (42.36), becoming almost identical to that of the base ASR model (46.15). All in all, our solution achieves the same person name accuracy as an ASR base model without sacrificing translation quality compared to a base ST system.

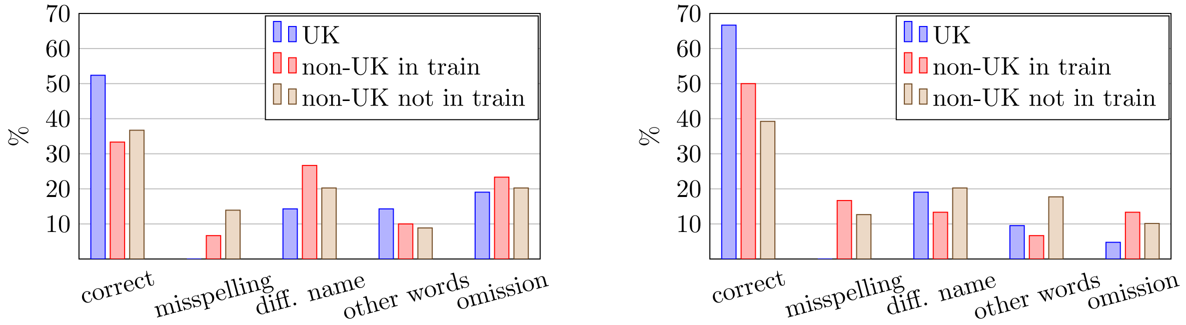
6.4.4 Error Analysis

While the goal is the correct rendering of person names, not all the errors have the same weight. For interpreters, for instance, minor misspellings of a name may not be problematic, an omission can be seen as a lack of help, but the generation of a wrong name is harmful, as potentially distracting and/or confusing. To delve into these aspects, we first carried out a manual analysis on the ASR outputs, and then compared the findings with the same analysis on ST outputs.

ASR Analysis

Two researchers with at least C1 English knowledge and linguistic background annotated each error assigning it to a category.[17] The categories, chosen by analyzing the system outputs, are: **misspelling** – when a person

[17] The inter-annotator agreement on label assignments was calculated using the *kappa coefficient* in Scott's π formulation (Scott, 1955), and resulted in 87.5%, which means "almost perfect" agreement in the standard interpretation (Landis and Koch, 1977).



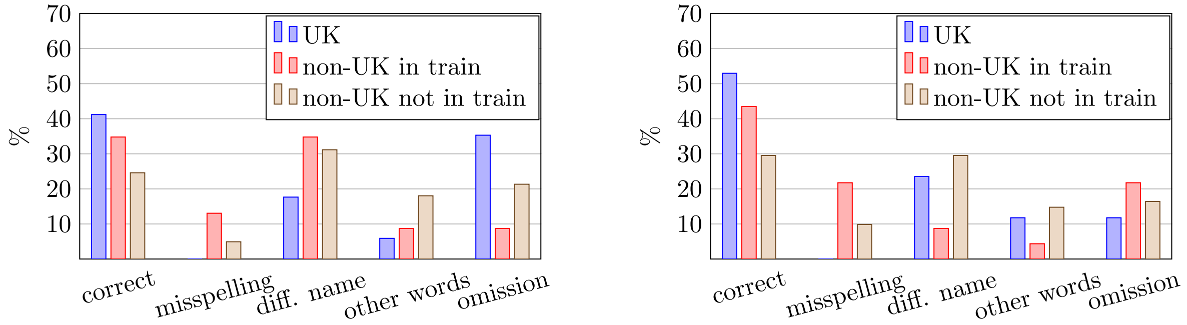
(a) Monolingual ASR errors.

(b) Multilingual ASR errors.

Figure 6.2: Correct person names and the categories of errors of the baseline and multilingual ASR systems.

name contains minor errors leading to similar pronunciation (e.g. *Kozulin* instead of *Kazulin*); **replacement with a different name** – when a person name is replaced with a completely different one in terms of spelling and/or pronunciation (e.g. *Mr Muhammadi* instead of *Mr Allister*); **replacement with other words** – when a proper person name is replaced by a common noun, other parts of speech, and/or proper nouns that do not refer to people, such as geographical names (e.g. *English Tibetan core* instead of *Ingrid Betancourt*); **omission** – when a person name, or part of a sentence containing it, is ignored by the system.

The results of the annotations are summarized in the graphs in Figure 6.2. Looking at the baseline monolingual system (Figure 6.2a), we notice that omissions and replacements with a different name are the most common errors, closely followed by replacements with other words, although for non-UK names the number of misspellings is also significant. The multilingual system (Figure 6.2b) does not only show a higher percentage of correct names, but also a different distribution of errors, in particular for the names belonging to the languages added to the training set (non-UK in train). Indeed, the misspellings increase to the detriment of omissions and replacements with a different name and other words. Omissions also



(a) Monolingual en-it ST errors.

(b) Multilingual ST *-it errors.

Figure 6.3: Correct person names and the categories of errors of the baseline and multilingual ST-into-Italian systems.

decrease for UK names and for names in languages not included in the training set (non-UK not in train). For UK names, the previously-missing names fall either into the correct names or into the replacements with a different name; for the non-UK not in train, instead, they are replaced by different names or other words.

Considering multilingual outputs, we observe that, for the languages in the training set (including English), in 66% of the cases the system generates a name that could be helpful for an interpreter (either correct or with minor misspellings). Confusing/distracting outputs (i.e., replacements with a different person name) occur in about 15% of the cases. Moreover, we notice that the system is able to discern when a person name should be generated (either correct, misspelled, or replaced by a different name) in more than 80% of the cases. This indicates their overall good capability to recognize patterns and/or appropriate contexts in which a person name should occur.

ST Analysis

The same analysis was carried out for ST systems translating into Italian (see Figure 6.3) by two native speakers. Although results are lower in

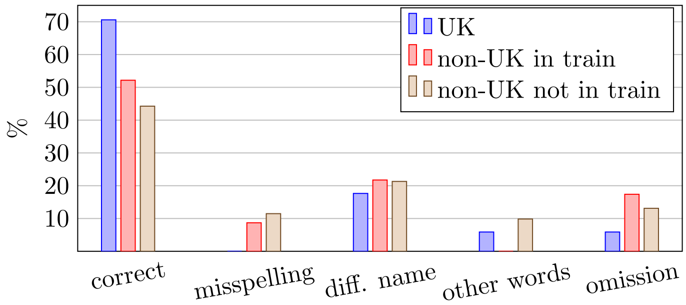


Figure 6.4: Correct person names and the different categories of errors of the ST-into-Italian triangle system with λ_{ASR}=0.8, λ_{ST}=0.2 expressed in percentages.

general, when moving from the monolingual (Figure 6.3a) to the multilingual (Figure 6.3b) system we can see similar trends to ASR, with the number of omissions and replacements with a different name that decreases in favor of a higher number of correct names and misspellings. Looking at the analysis of the triangle model with λ_{ASR}=0.8, λ_{ST}=0.2 presented in §6.4.3 (Figure 6.4), we observe that misspellings, omissions, and replacements with other words diminish, while correct names increase. Moreover, both the accuracy (i.e., *correct* in the graphs) and the error distributions of this system are similar to those of the ASR multilingual model (Figure 6.2b). On one side, this brings to similar conclusions, i.e. ST models can support interpreters in ∼66% of the cases, and can discern when a person name is required in the translation in ∼80% of the cases. On the other, it confirms that the gap with the ASR system is closed, as observed in §6.4.3.

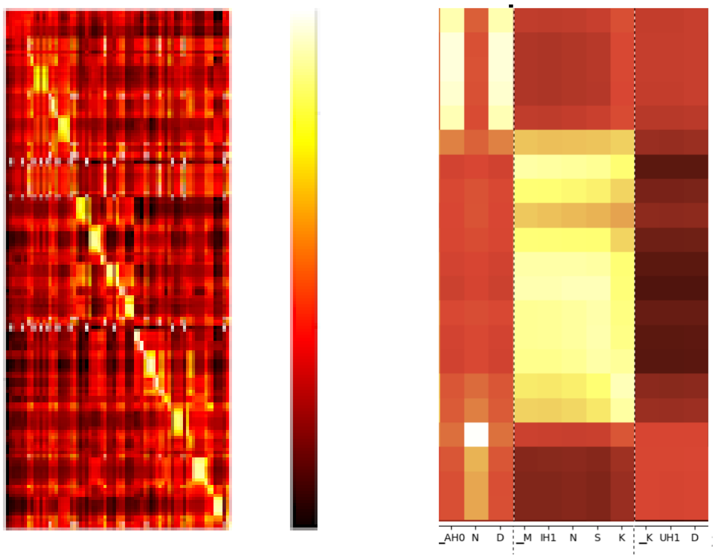
6.4.5 Summary

The analysis of §6.3 revealed that ST systems struggle in handling person names. In this section, we demonstrated that the problem mostly comes from names that are unknown to systems and in languages they have not been trained on. Therefore, we designed dedicated architectural solutions aimed to improve the ability to handle specific elements. Along this line of research, we have shown that a multilingual triangle ST model, which gives more weight to the transcription than to the translation task, has relative improvements in person name accuracy by 48% on average. We also acknowledge that much work is still needed in this area, with a large room for improvement still available, especially to avoid the two most common types of errors unveiled by our analysis: omissions and replacements with different person names. In the next section, we investigate solutions to integrate external knowledge (in the form of a dictionary of names likely to appear in a given domain), with the goal of further increasing the correct recognition and spelling of the names.

6.5 Named Entity Detection and Injection

As seen in §6.2.1 and §6.2.3, since neural translation systems are known to struggle in presence of rare words (Koehn and Knowles, 2017), a category to which many NEs belong, researchers studied dedicated solutions that exploit additional, contextual information (in the form of lists or dictionaries of domain-specific words/phrases) available at inference time both in ASR and MT. No work, however, targeted the ST scenario, where the main problem is assessing which entries of the dictionary are relevant for an utterance. Here, the pattern matching approach used in text-to-text (T2T) translation is not feasible because of the different input modality, and ASR solutions are not directly applicable as they do not deal with the language switch between the source and target.

Motivated by the difficulties of ST systems in correctly handling NEs shown in the previous sections, the practical relevance of the problem, and the lack of existing solutions, in this section we present the first approach to exploit contextual information – in the form of a bilingual dictionary of NEs – in direct ST. Specifically, our main focus is the detection of the



(a) Full sentence. (b) Zoom on *Minsk* phonemes

Figure 6.5: Heatmap of cosine similarities (the lighter, the more similar) between the encoder outputs of text and speech of the ST2T model released by Tang et al. (2021). On the x axis, each item is a phoneme passed to the textual shared encoder; on the y axes there are frames that correspond to the utterance. The full sentence is: *Madam President, the resolution on the situation in Belarus reveals what Brussels and Minsk could do in order not to lose the momentum for improving their relations.*

NEs present in an utterance, among those in a given contextual dictionary. Performing this task allows us to rely on existing solutions for ASR (see §6.2.3) to inject the correct translations for the NEs. In particular, we propose the adoption of a decoder architecture similar to CLAS (see §6.2.3) and provide it with the list of translated NEs considered present by our detector module. Experimental results on NEuRoparl-ST demonstrate that we can improve NE accuracy by up to 7.1% over a base ST model, and reduce the errors on person names by up to 31.3% with respect to a strong baseline exploiting the same inference-time contextual data.

6.5.1 Entity Detection for ST

Two operations are necessary to exploit a dictionary of NEs likely to appear in an utterance: *i)* detect the relevant NEs among those in the dictionary, *ii)* look at the corresponding translations to accurately generate them. Accordingly, we add two modules to the ST model: *i)* a detector identifying the NEs present in the utterance, and *ii)* a module informing the decoder about the forms of the detected NEs in the target language.

A recent research direction in ST consists in training models that jointly perform ST and MT to improve the quality of direct ST (Tang et al., 2021; Ye et al., 2022). These speech/text-to-text (ST2T) models include auxiliary tasks to force the encoder outputs of different modalities to be close when the text/audio content is the same. Figure 6.5 confirms that encoder outputs for text (the text is actually converted into phonemes before being fed to the encoder, as per Tang et al. 2021) and audio are indeed similar. Specifically, there is a strong similarity between the phonemes that compose a word and the audio frames that correspond to that word. Based on this, we hypothesize that we can use the encoded representation of the textual NEs in a dictionary/list and the encoded representation of an utterance to determine whether each NE has been mentioned or not.

Similarity-based Detector

Following the above considerations, a first naive attempt to detect a NE in the audio is to look whether along the audio dimension (the y axis) there is an instant with high similarity with the phonemes corresponding to the NE. However, since the similarities are at word level, we would be unable to detect NEs with more than one word, which is actually very frequent. Hence, we defined a word-based approach to obtain a likelihood score for the presence of an entity in an utterance. For each word of a NE, we look for a region in the audio with high similarity and we want this region to span over many consecutive audio frames to avoid spurious local similarities. Then, we repeat the same operation for the next word in the NE, where the starting frame (time) of this word is the last frame (end time) of the previous word. Finally, we average the scores of all the words that compose the NE and take the maximum score we can achieve.

Although we noticed significant improvements over the first naive method, the precision of this solution still resulted very low, with many false positives. As such, we decided to look for alternatives, moving to an approach based on a trained detector.

Trained Detector

A trained detector is a neural network that estimates the probability that a given NE is present in an utterance or not.

At training time, we feed a positive sample (i.e., a piece of text actually present in the utterance) and a negative sample (i.e., a piece of text not present in the utterance) for each audio to train the NE detector. Positive and negative texts can be sampled *i)* from random words in the transcript of the current utterance and of those of other utterances in the same batch,[18] or *ii)* from automatically-detected NEs. The second method is closer to the real goal but limits the amount of training data (ignoring the utterances in the training set that do not contain NEs), and its variety (at the risk of overfitting to the NEs in the training set). To avoid this, we adopt a mixed approach, where in training samples without NEs the first method is used, while in training samples with NEs one of the two methods is randomly selected (assigning 80% of probability to choose automatically-recognized NEs).

Another critical aspect is the design of this detector module. We first

[18] Ensuring they are not present in the examined utterance.

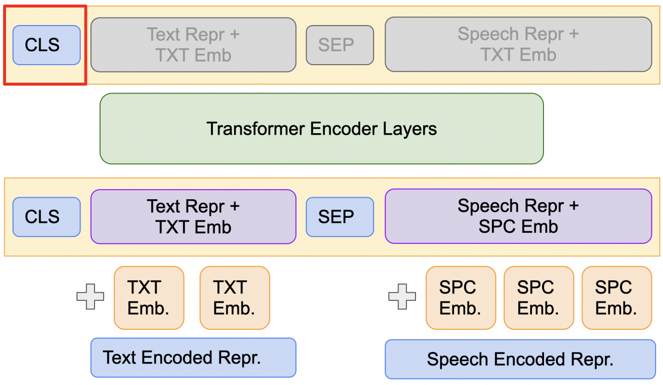


Figure 6.6: NE Detector architecture.

tested the Speech2Slot architecture (Wang et al., 2021b), a stack of three layers made of a multi-head attention (MHA) followed by an FFN without residual connections. The NE textual encoder output is fed as query to the MHA, while the key and values are built from the speech encoder output. Unfortunately, training this architecture turned out challenging and the network failed to converge.

In light of this, we resorted to a stack of three Transformer encoder layers, fed with a concatenation of a *CLS* token, the NE textual encoder output, a *SEP* token, and the utterance ST2T encoder output. From the output of the last layer, we then select only the first vector, corresponding to the *CLS* token, and feed it to a *sigmoid* (σ) activation function to get the probability that the NE is present in the utterance. In addition, we add a trained *TXT* embedding to all the NE textual encoder vectors and a trained *SPC* embedding to all the speech encoder vectors, obtaining the architecture represented in Figure 6.6. Lastly, as a NE should appear in a contiguous and relatively short speech segment, we force the module to focus on a limited span of speech vectors surrounding the considered one by means of an *attention masking* mechanism. Specifically, as the amount of speech that should be considered depends on the NE length, we mask all the

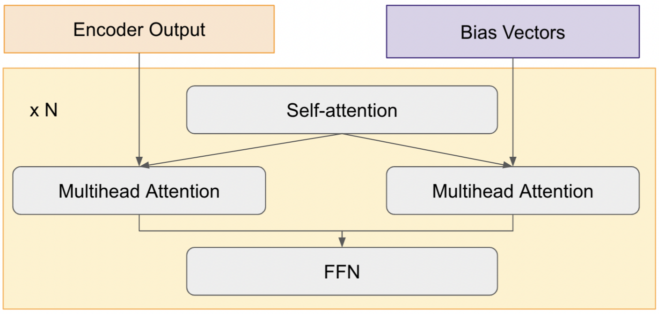


Figure 6.7: CLAS Transformer decoder layer (*parallel* method).

speech vectors that are farther than two[19] times the number of phonemes of the NE to detect with respect to the current speech element. For instance, when trying to detect a NE made of 10 phonemes, each speech vector can attend only to itself, the 20 speech vectors before it, and the 20 after it, in addition to the textual and the special token vectors. In other words, each speech vector can attend to the surrounding ones, to all textual vectors, and the *CLS* and *SEP* embeddings.

6.5.2 Decoding with Contextual Entities

As we will see in §6.5.4, the entity detector achieves high recall, but false positives are hard to avoid. Hence, to demonstrate that our entity detector is useful despite a low precision, we inject the selected entities into the model with an approach tolerant to false positives. We adopt an architecture similar to CLAS (see §6.2.3 – Pundak et al. 2018), where the bias encoder is a trained 3-layer Transformer encoder, and the attention between the decoder and the bias encoder outputs is an MHA implemented following the *parallel* or *sequential* methods described in §4.3.2 (see Figure 6.7). Each NE in the list of those considered likely present (*bias-NE*) is encoded with the bias encoder and the encoder outputs are averaged to get a single vector. After repeating this step for all the bias-NEs, the resulting vectors are

[19] We also tested 1 and 3, noticing minimal differences and chose 2 due to its lower loss on the dev set.

concatenated together with a *no-bias* learned vector that allows our model to ignore the information from bias vectors.

6.5.3 Experimental Settings

For our experiments, we build a multilingual direct ST2T and a multilingual cascade system (ASR followed by multilingual MT), following the recipe in (Tang et al., 2022).

Data and Evaluation

As monolingual text data for the pre-training of the direct ST2T and ASR models, and as parallel text data for MT, we use Europarl.[20] As unsupervised audio data, instead, we use Libri-Light (Kahn et al., 2020). Finally, as supervised data, we use MuST-C and Europarl-ST. In particular, for ASR we use their en→es section (the largest language direction among the ones we considered), while for ST we use the en→{es,fr,it} sections.

The quality of the NE detectors is assessed in terms of the trade-off between recall and the number of NEs retrieved. We estimate selectivity through the number of NEs retrieved instead of precision, as some NEs may be correctly detected even though they are not annotated in NEuRoparl-ST,[21] making hard to reliably compute precision. The output of the ST systems is evaluated with SacreBLEU[22] on Europarl-ST for the translation quality, and with case-sensitive entity accuracy on NEuRoparl-ST for the ability in handling NEs. Among NEs, we focus on geopolitical entities (GPE), locations (LOC), and person (PER) names, as these three types are the most challenging for ST systems (see §6.3).

[20] We filter from Europarl all the data that belongs to the dates of the talks inside the Europarl-ST test set, and therefore also to NEuRoparl-ST, which is derived from it.

[21] For instance, this happens when a NE is part of a bigger one (e.g. the NE *Lisbon* is retrieved in a sentence that contains the NE *Treaty of Lisbon*).

[22] `case:mixed|eff:no|tok:13a|smooth:exp|v:2.1.0`

Architecture and Training Details

Our ASR and ST2T models directly process raw waveforms using the same hyperparameters of (Tang et al., 2022). Encoder layers are randomly dropped (LayerDrop) at training time with 0.1 probability (Fan et al., 2019). The multilingual MT models have 6 encoder and decoder layers with 512 features and 1024 FFN hidden features. When training CLAS models, we initialize the weights with those of the pre-trained ST2T model. We freeze encoder weights, and we also experimented with freezing the decoder parameters from the pre-trained ST2T model: in this case, we train only the newly added components and the output projection layer.

For the training of our ASR and ST2T systems, we first perform a BART text pre-training on monolingual text data, followed by joint pre-training that also includes the unsupervised and supervised audio data. The resulting model is the base for both our ASR and ST2T models. While the ASR system is fine-tuned only on the ASR data, the ST2T is fine-tuned on both the ST corpora and the ASR data, although the auxiliary ASR task is not used at inference time. All the trainings have been performed on 8 V100 GPUs, using the batch sizes indicated by (Tang et al., 2022).

6.5.4 Results

NE Detection

Table 6.13 reports the retrieval results of the trained NE detector module described in §6.5.1, isolating the contribution of its components, and compares it with the algorithm based on the cosine similarities, which is unable to obtain good selectivity. First, we notice that, to achieve meaningful scores, it is essential to introduce LayerDrop when extracting the input features using the shared speech/text encoder of the ST2T model. Otherwise, the results are close to a random predictor. Speech masking

	GPE	**LOC**	**PER**	**Retr.**
Cosine Similarities	68.2%	74.3%	53.2%	138.2
Base NE detector	31.4%	6.3%	28.3%	115.5
+ speech masking	31.9%	17.9%	29.4%	54.3
+ layerdrop	57.2%	24.2%	38.0%	3.9
+ layerdrop	66.8%	33.7%	40.2%	4.3
+ train on NE	93.9%	76.8%	79.4%	1.8
+ modality emb.	95.2%	94.7%	78.3%	1.8
+ attn. masking	93.5%	93.7%	90.2%	1.6
+ max word len 5	96.5%	93.7%	89.1%	1.4
+ margin ranking	96.1%	91.6%	88.0%	1.2

Table 6.13: Recalls on GPE, LOC, and PER, and average number of NEs retrieved per utterance (Retr.). For each utterance, the NE detectors are fed with all the distinct GPE, LOC, PER, and organizations (ORG) in the test set for a total of 294 NEs. A NE is considered detected if the NE detector assigns a detection probability higher than 86%.

also helps, but is harmful when combined with LayerDrop. Moreover, feeding automatically-detected NEs at training time with the mixed approach described in §6.5.1, instead of only using random words, greatly improves both recalls and selectivity. The addition of trained modality embeddings also proved helpful, especially for LOC and GPE recall. The attention masking provides significant benefits in terms of PER recall and selectivity, at the cost of a very limited degradation on GPE and LOC recall. Further improvements in selectivity were obtained by picking more than a single random word when training the NE detector (up to 5 consecutive words), and by adding an auxiliary margin ranking loss to the binary cross entropy loss. This final module achieves recalls higher or close to 90%, retrieving 1.2 NEs per utterance on average (the test set contains 0.34 NEs from these 3 categories on average). Excluding the retrieved NEs present in an utterance but not annotated as such in the test set,[21] we can compute the precision of this module, which is 55.8%. The non-negligible number of false positives is investigated in §6.5.5, and highlights that the NE detector can be used to create a short-list of NEs likely present in the sentence,

	BLEU	**GPE**	**LOC**	**PER**	**Avg.**
Cascade	37.6	80.0	74.2	51.2	68.5
Base ST2T	**38.8**	82.2	78.4	49.3	70.0
+ CLM (λ=0.10)	**38.8**	83.9	76.8	50.8	70.5
+ CLM (λ=0.15)	38.0	83.6	74.9	52.7	70.4
+ CLM (λ=0.20)	37.0	82.5	73.0	53.4	69.6
Parallel CLAS	37.5	**84.7**	78.4	66.1	76.4
+ freeze decoder	37.0	82.8	78.7	64.6	75.4
Sequential CLAS	35.8	84.5	78.7	**68.0**	**77.1**
+ freeze decoder	36.8	82.7	**79.9**	**68.0**	76.9

Table 6.14: Translation quality (BLEU) and accuracy for GPE, LOC, and PER – as well as the average over the 3 categories (Avg.) – of the base direct ST2T, cascade, and the test-entities aware systems (class LM – CLM – and CLAS models). The results are the average over the 3 language pairs (en→es,fr,it).

rather than to enforce the presence of detected NEs, motivating the CLAS solution (§6.5.2).

ST Quality and NE Translation

Our CLAS method leverages additional data available at inference time. Its comparison with a plain ST model would hence be unfair, so we introduce a strong baseline that exploits this additional data. Specifically, we perform a class-based LM rescoring of the ST model probabilities (see §6.2.3), using shallow fusion (i.e., adding to the ST model probabilities the LM probabilities rescored with a weight λ). We train the class LM on the test-time NEs, and a generic LM on the target side of the MT training data. At each decoding step, if we are inside NE tags for the current hypothesis, we rescore (shallow fusion) the ST outputs with the class LM; otherwise, the rescoring is done with the generic LM.

Table 6.14 compares this strong baseline, the base model, and our CLAS systems fed with the entities selected by our NE detector module. We can see that CLAS systems are the best in NE accuracy, reducing by up to 31% the number of errors for person names compared to the best baseline using the same additional information. The improvements for other NEs are lower: we argue that the reason lies in the different textual representations NEs have in the different languages (source and target) while person names are mostly the same. Despite its better NE handling, CLAS suffers from a 1.3-2.0 BLEU degradation with respect to the baseline. However, comparing our Parallel CLAS model to the baselines, we notice that the best baseline for person names (CLM λ=0.20) is significantly inferior on all metrics, including BLEU and person name accuracy. Moreover, BLEU is similar to the cascade solution with significantly higher accuracy on all NE categories.

6.5.5 Analysis

As observed in §6.5.4, the weakness of the NE detector is the number of wrongly detected NEs (false positives). To better understand why they happen, we conducted a manual analysis of the false positives, assigning each of them to one of the following categories: *i)* **similar semantic** (13.7%), NEs detected in an utterance where there is a NE with a similar meaning (e.g. *Chamber/Parliament*) or there is another NE of the same type (e.g. *Pakistan/Afghanistan*); *ii)* **similar phonetic** (14.3%), NEs detected in sentences where there is a word that is similar or sounds similar (e.g. *President/Presidency*); *iii)* **partial match** (34.0%), NEs detected in utterance where only part of the NE is present (e.g. *Fisheries Committee/Budget Committee*); *iv)* **acronyms** (8.4%), these NEs are poorly handled because our text-to-phonemes converter does not handle them properly (e.g. *US* is converted as the pronoun *us* and *EU* as the pronoun *you*); *v)* **different form** (16.5%), NEs detected where the same NE is mentioned but in a different form (e.g. *government of Malaysia/Malaysian Government*), so these are not real errors; *vi)* **uninterpretable** (13.1%), the human cannot understand the reason for the error. This inspection shows that future work should focus on training strategies that alleviate the detection errors of similar words and partial matches, creating systems more robust to small yet significant variations between different entities.

6.5.6 Summary

After showing the weaknesses of ST systems in handling NEs, especially person names (§6.3), and investigating the causes of low person name accuracy with dedicated mitigations (§6.4), in this section we explored how to leverage dictionaries of NEs in a specific domain/context to improve the NE translation accuracy of ST models. We mainly focused on the detection of which NEs of a domain dictionary are present in the input utterance, proposing an additional module on top of the encoder outputs that determines whether each NE is present or not. Such a module achieved a high recall for geopolitical entities, locations, and person names, while the most prominent challenge regards increasing the selectivity of the model. In addition, our thorough analysis of the most common categories of false positives allowed us to identify guidelines and promising directions for future works on the topic. Lastly, we proposed a method to inject the selected NEs in the decoding phase, showing that the proposed detection strategy is already capable of improving NE handling, with average accuracy gains of up to 14.4% on GPE, LOC, and PER over strong baselines leveraging the same inference-time information. The next section moves another step, complementary to the improvement of NE accuracy, toward augmented ST systems by exploring a joint execution of the ST and the NER tasks with a single model.

6.6 Joint Translation and Named Entity Recognition

The augmented ST paradigm requires not only an accurate translation of relevant entities (covered so far in this chapter), but also supporting the users by highlighting important elements (such as NEs) and eventually providing related information. In this framework, a critical task is the output enrichment with information regarding the mentioned entities. This is currently achieved by (post-)processing the generated translations with NER tools and eventually retrieving their description from knowledge bases (an aspect we do not cover in this work). In light of the recent promising results shown by direct systems and the known weaknesses of cascades (error propagation and additional latency), we explore multitask models that jointly perform ST and NER, and compare them with a cascade baseline that performs the two tasks sequentially. In doing so, we address the following research questions: *is the current cascade of ST and NER the best approach? What are the effects of jointly performing the two tasks on NE accuracy and translation quality?*

Our experiments on NEuRoparl-ST demonstrate that joint models significantly outperform the ST+NER cascade by 0.4-1.0 F1 in the NER task, while being on par in terms of translation quality. Such improvement is achieved without introducing any significant computational overhead with respect to a plain ST model, thus being remarkably more efficient than cascade systems.

6.6.1 Joint NER and ST

The easiest way to extract the NEs from a translation consists in applying a NER model on the output of the ST model. Henceforth, we refer to this approach as *cascade*, and we consider it as a baseline for comparison against our systems that jointly perform the two tasks with a single model. Our solutions – *inline*, and *parallel* – are described below:

Inline (Fig. 6.8). The vocabulary of the direct ST model is extended with tags that represent the start (e.g., `<LOC>`) and end (e.g., `</LOC>`) of the

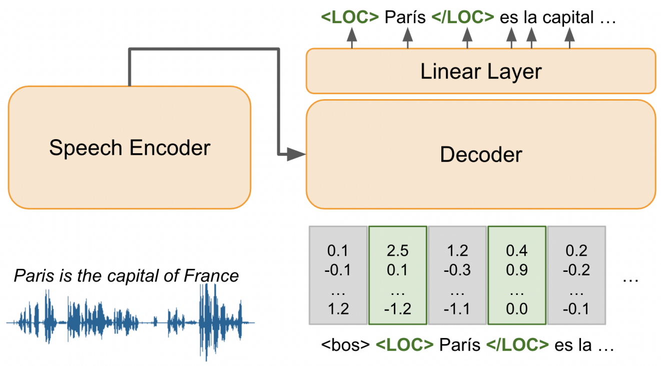


Figure 6.8: Architecture of the *inline* solution. The additional tokens generated in the output are highlighted in green, and are passed to the decoder as all the other previous output tokens.

NE categories to be recognized which, in our case, are 18.[23] These tags are treated as any other token (or subword): they are predicted in the output sequence, and – together with the other tokens – fed to the decoder as previous output tokens, informing it about the NE categories. This solution does not require any architectural changes to the ST model but introduces additional overhead, especially at inference time, as the higher number of tokens to generate (due to the additional start/end NE tags) leads to an increase in the number of forward passes on the autoregressive decoder.

Parallel (Fig. 6.9). At each time step, the output of the last decoder layer is processed in parallel by two linear layers. The first linear layer maps the vectors to the vocabulary space to predict the next token as in standard ST models. The second linear layer maps the same vector to the NE-category space to predict the NE category to which the token belongs to, if any, or O (i.e., $OTHER$), if the token is not part of a NE. Although the second linear layer introduces additional parameters to train, its computational cost is negligible compared to that of the whole decoder. Moreover, this solution avoids the supplementary decoder forward passes required by the

[23] The categories are those defined in the OntoNotes annotation.

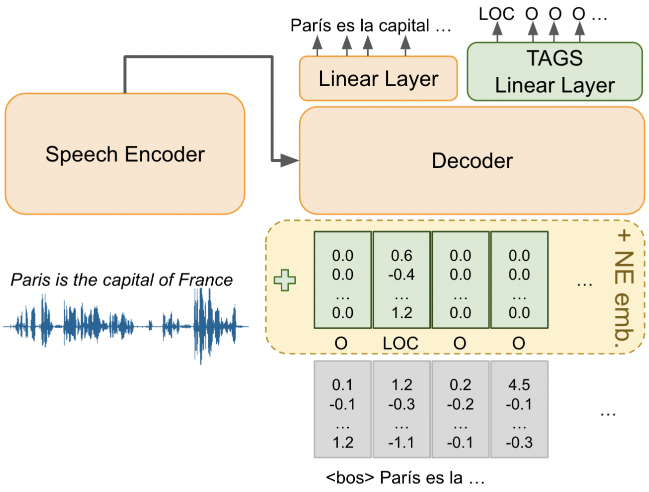


Figure 6.9: Architecture of the *parallel* solution. The introduced linear layer (in green) is processed token-by-token in parallel with the other linear layer. In the *+ NE emb.* variant (yellow dotted area), the previously predicted tags are converted into embeddings that are summed to the corresponding previously generated tokens.

inline method. However, the potential drawback in comparison with the inline solution is that it cannot exploit information about the NE categories assigned to the previously generated tokens during translation. As we posit that this lack of information may cause performance degradations, we propose a variant of this method, in which the embeddings of the previous output tokens are summed with learned embeddings of their corresponding NE categories.[24] This change requires only 19 additional embeddings to learn (one for each NE category, plus O) – a negligible number compared to the target vocabulary size – and a sum, hence producing no significant computational overhead. Note that the inline solution requires 36 additional embeddings instead of 19 as in this case, since it differentiates between the start and the end of each NE category. We refer to this variant as **Parallel + NE emb.** (**Fig. 6.9**).

[24] The beginning-of-sentence (*bos*) token is considered of O category.

6.6.2 Experimental Settings

Data and Evaluation Metrics

All models are trained on the MuST-C and Europarl-ST corpora. To train the joint NER and ST models, we automatically annotated the NEs on the target translations with the same NER tool used in our cascade approach, obtaining parallel training data with speech and the corresponding annotated translations without any manual intervention. Translation quality is evaluated with SacreBLEU on the Europarl-ST test set. The capability to correctly render/translate NEs is measured on the NEuRoparl-ST benchmark with NE accuracy (case-insensitive, for the sake of comparison with the results in §6.3), and with F1, which also measures the ability to recognize NEs. The F1 is computed considering as correct only those NEs that are accurately translated and identified (regardless of the category they have been assigned to). As such, NEs poorly translated and recognized by a model penalize both recall and precision. This strict definition of F1 mirrors the needs dictated by users' perception: in augmented ST, while unrecognized NEs are only a lack of help to the users, recognized but spurious NEs are more harmful as they would distract them with unrelated and potentially misleading content. Finally, the ability to assign a NE to the correct category is evaluated with accuracy, i.e. the percentage of NEs assigned to the correct category.

Architectures and Training Details

All our ST models have a Conformer encoder and Transformer decoder, with the same settings described in §3.6.3. We optimize the label-smoothed cross-entropy loss with 0.1 smoothing factor with Adam, and the learning rate is initially increased for 20k steps up to $5 * 10^{-3}$, and then it decreases with inverse squared root policy. Our mini-batches contain 10,000 tokens,

	en-es				**en-fr**				**en-it**			
	BLEU	**NE Acc**	**F1**	**Cat. Acc.**	**BLEU**	**NE Acc**	**F1**	**Cat. Acc.**	**BLEU**	**NE Acc**	**F1**	**Cat. Acc.**
Best §6.3	37.7	71.4	-	-	30.1	67.3	-	-	26.0	67.3	-	-
Cascade	37.9	71.9	49.1	89.8	36.2	69.2	44.8	90.2	28.3	66.5	44.5	88.8
Inline	37.9	72.2	$49.5^{\dagger\ddagger}$	90.1	36.3	69.6	$45.6^{\dagger\ddagger}$	90.2	28.3	66.9	$45.5^{\dagger\ddagger}$	89.4
Parallel	38.1	71.9	48.1	89.5	36.1	69.0	44.5	90.6	28.4	67.5	43.9	89.1
+ NE emb.	38.0	72.1	$49.5^{\dagger\ddagger}$	89.9	36.1	69.3	$45.5^{\dagger\ddagger}$	90.4	28.2	67.3	$45.4^{\dagger\ddagger}$	89.1

Table 6.15: SacreBLEU, case-insensitive NE accuracy, F1, and category assignment accuracy (Cat. Acc.) of previous work, our cascade of ST and NER, and the proposed joint ST+NER models. All results are the average of three runs. † indicates statistically significant improvements over *cascade*, and ‡ over *parallel*. A result is considered statistically significant when we can reject with 95% confidence the null hypothesis that the considered mean is not higher than the mean of the baseline (Student, 1908).

we set to 8 the update frequency, and train on 4 K80 GPUs. We stop the training after 10 epochs without loss decrease on the validation set, and average 5 checkpoints around the best. As NER system, instead, we rely on a multilingual BERT-based model,[25] openly-available in DeepPavlov.

6.6.3 Results

Table 6.15 compares our cascade baseline, the joint NER+ST inline and parallel methods, and the best results on the same benchmark reported in §6.3 with a system trained on a large amount of data.

First of all, we can notice that, even though trained on fewer data, all our systems outperform the previous ones both in terms of translation quality (BLEU), and NE accuracy. The difference is remarkable on en-fr, where our systems are ∼6 BLEU better, while scores are on par on NE accuracy only in the en-it direction. This is likely motivated by the different and improved architecture of the models and confirms the soundness of our experimental settings, as well as the strength of the cascade baselines and the reliability of our results.

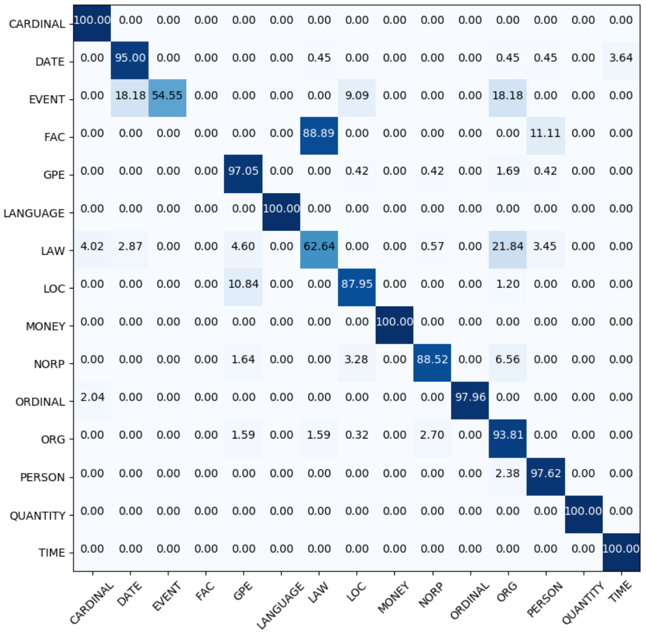
Looking at translation quality – both generic (BLEU) and specific to

[25] `http://docs.deeppavlov.ai/en/master/features/models/bert.html`

NEs (NE accuracy) – we see that all methods (cascade and joint) achieve similar results. The small differences among the scores (0.2 BLEU and up to 0.6 NE accuracy) are not consistent across language directions and are never statistically significant, thus being ascribable to fluctuations due to the inherent randomness of neural methods. We can conclude that the additional NER task does not help to improve NE translation (in contrast with previous findings for MT, see Xie et al. 2022), but it also does not degrade generic translation quality, as it could have happened since part of the model capabilities has to be dedicated to the additional task.

When we consider the F1 metric, instead, the results highlight the differences between the various approaches. Our joint NER and ST models beat the cascade by a significant margin (0.4-1.0 F1). This is surprising if we consider that the training data of the joint methods was generated with the NER system of the cascade approach, and highlights the strength of direct multitask systems. Among the joint solutions, the *inline* and *parallel + NE emb.* significantly outperform the *parallel* method, demonstrating the importance of providing the decoder with information about the NE category assigned to the previously generated tokens. The difference between *inline* and *parallel + NE emb.* is instead very limited (0.1, if any) in favor of the *inline*, and is not statistically significant. These two methods can therefore be considered on par.

Lastly, all systems show a good ability in NE category assignment. The accuracy differences range between 0.6 and 0.3, are not coherent across language pairs, and are never statistically significant. Not only the overall performance of the systems is on par, but also their confusion matrices over the NE categories are basically the same on all language pairs. As an example, Figure 6.10 reports the confusion matrix of the *parallel + NE emb.* model for en-es. We notice that the categories that are more difficult to recognize for our models are facilities (FAC), events ($EVENT$), and



	CARDINAL	DATE	EVENT	FAC	GPE	LANGUAGE	LAW	LOC	MONEY	NORP	ORDINAL	ORG	PERSON	QUANTITY	TIME
CARDINAL	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DATE	0.00	95.00	0.00	0.00	0.00	0.00	0.45	0.00	0.00	0.00	0.00	0.45	0.45	0.00	3.64
EVENT	0.00	18.18	54.55	0.00	0.00	0.00	0.00	9.09	0.00	0.00	0.00	18.18	0.00	0.00	0.00
FAC	0.00	0.00	0.00	0.00	0.00	0.00	88.89	0.00	0.00	0.00	0.00	0.00	11.11	0.00	0.00
GPE	0.00	0.00	0.00	0.00	97.05	0.00	0.00	0.42	0.00	0.42	0.00	1.69	0.42	0.00	0.00
LANGUAGE	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LAW	4.02	2.87	0.00	0.00	4.60	0.00	62.64	0.00	0.00	0.57	0.00	21.84	3.45	0.00	0.00
LOC	0.00	0.00	0.00	0.00	10.84	0.00	0.00	87.95	0.00	0.00	0.00	1.20	0.00	0.00	0.00
MONEY	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
NORP	0.00	0.00	0.00	0.00	1.64	0.00	0.00	3.28	0.00	88.52	0.00	6.56	0.00	0.00	0.00
ORDINAL	2.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	97.96	0.00	0.00	0.00	0.00
ORG	0.00	0.00	0.00	0.00	1.59	0.00	1.59	0.32	0.00	2.70	0.00	93.81	0.00	0.00	0.00
PERSON	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.38	97.62	0.00	0.00
QUANTITY	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
TIME	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Figure 6.10: Confusion matrix over the 15 NE categories with at least one NE correctly translated and recognized for the *parallel + NE emb.* system on en-es. On the y-axis there are the true labels, while on the x-axis the predicted labels. The numbers are percentages computed on the y-axis.

names of laws (*LAW*), while all the other categories achieve high accuracy. Among the three critical ones, *FAC* and *EVENT* are very rare (19 and 9 occurrences in the test set), while *LAW* is more frequent (141 occurrences), thus representing the main source of assignment errors. The root of this difficulty may lay in the nature of laws, which have high variability, are long, and frequent only in specific domains. At last, another common source of errors is labeling location names as *GPE*, which is understandable as their categorization is highly dependent on the context in which they occur (e.g. *Europe* as a continent is a *LOC*, but in politics it can also refer to a *GPE*).

6.6.4 Efficiency

One known advantage of direct systems over the cascade ones is their lower overall computational cost since they need a forward pass on only one model

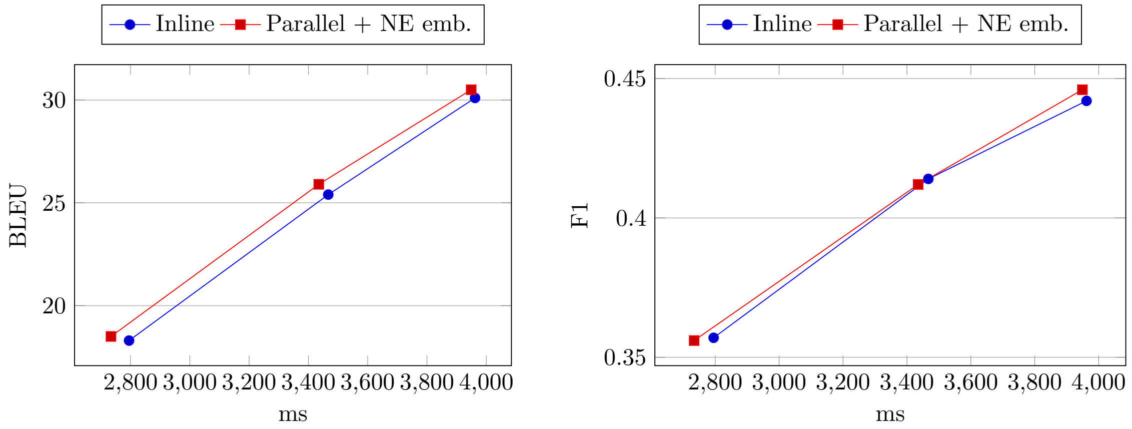


Figure 6.11: BLEU-LAAL, and F1-LAAL curves of the *inline* and *parallel + NE emb* for k=1,2,3 in en-es. All points are the average over three runs.

instead of two. However, the computational cost of the two proposed joint solutions is different as well, as the number of decoding steps (i.e., forward passes on the autoregressive decoder) is different. Indeed, the *inline* method has to predict the start and end NE tags, requiring on average 7% more decoding steps (in the test set) compared to a plain ST model and to the *parallel* system, which does not introduce additional decoding steps.

The computational cost is particularly critical if the translation has to be generated in real-time, i.e. in simultaneous ST, where it directly affects the output latency. For this reason, we conclude our work by comparing the two best models (*inline* and *parallel + NE emb*) in the simultaneous setting using the popular wait-k (Ma et al., 2020) policy. This allows us to estimate the overhead introduced by the additional decoding steps of the *inline* model. In Figure 6.11, we report the BLEU-latency, and F1-latency curves of the two models on en-es (the curves for en-fr and en-it show the same trends), where the latency is measured through computational-aware length-adaptive average lagging (LAAL – Papi et al. 2022). The two curves show that the *parallel + NE emb* model has a slightly better trade-off thanks to its lower computational cost. However, since the computational cost only accounts for a fraction of the latency (∼53% of the computational-aware LAAL is due to the wait time of the wait-k policy), and the computational difference is not large (∼5%), the gap between the two models is limited.

All in all, we can conclude that the *inline* model introduces a computational overhead that depends on the number of NEs detected in an utterance. On our test set, with 1,267 sentences, 30.6K words, and 1,638 NEs, we estimated as 5% its computational overhead in time compared to a base direct ST model and to our *parallel + NE emb.* solution. In light of the similar quality of *inline* and *parallel + NE emb.* systems, this difference – which may be larger in domains where NEs are more frequent, as news or molecular biology (Nobata et al., 2000) – makes the *parallel + NE emb.* method our best solution overall.

6.6.5 Summary

As an additional step toward augmented ST, in this section we went beyond the analysis and improvement of NE translation quality (see §6.3, §6.4, §6.5) and presented the first multitask models jointly performing speech translation and named entity recognition. First, we showed the importance of properly feeding information about the previously predicted NE tags, as done in the *inline* and *parallel + NE emb.* models. Second, and most importantly, we showed that our joint solutions consistently outperform a cascade system on the NER task (by 0.4-1.0 F1), while being on par in terms of translation quality. Lastly, we evaluated the computational efficiency of our methods, demonstrating that the *parallel + NE emb.* system, which does not introduce noticeable overhead with respect to a plain ST model, is more efficient than the *inline* method, besides being on par in terms of translation and NER quality. As such, it represents the most attractive solution to perform the ST and NER tasks with high quality, at the cost of a single model.

6.7 Conclusions

Following the huge improvements in terms of translation quality (§3, §4), we investigated the strengths and weaknesses of direct ST models for their application in the "augmented translation" paradigm. In this scenario, the ST model should complement and "augment" humans by providing support that eases and speeds up their work. Concretely, this means that ST outputs should convey the correct information in particular with respect to mentioned entities, numbers, and specific terminology – which cause a high cognitive workload to translation professionals – and eventually provide them with contextual information. The task requires the identification of the mentioned entities and their linking with knowledge bases (although this second step is not covered in this thesis). Toward these goals, we started by creating NEuRoparl-ST, an extension of Europarl-ST, which is the first benchmark (for en→es,fr,it) openly available to assess the ability of ST systems in translating NEs and domain-specific terms. Then, we used it to compare the cascade and direct ST paradigms, demonstrating that they have similar capabilities and both struggle with person names. Following these findings, we identified the causes of errors on person names in the nationality of the referent and training frequency, and proposed a multilingual triangle model to mitigate the problem. In addition, as interpreters and translators often have access to dictionaries of NEs likely to appear in a given domain, we investigated the integration of such resources in direct ST models. To the best of our knowledge, our solution is the first direct ST system integrated with external dictionaries of NEs. Experiments with this solution show significant gains, outperforming by 14.4% in person-name accuracy other solutions borrowed from the ASR field and exploiting the same kind of data. We concluded our study of the topic by introducing the first models that jointly perform ST and NER: our best system has significantly higher NER accuracy than a pipeline of dedicated ST and NER models, and keeps the computational cost as low as a single base ST model. Overall, these contributions pave the way for the integration of direct ST models in augmented ST applications, albeit leaving ample room for improvement on the topic to pursue in future works, such as reducing the false positives when detecting the NEs mentioned in an utterance and improving its computational efficiency.

Chapter 7

Conclusion

When this PhD started, in November 2019, the main question within the ST community was: will direct ST models be able to keep their promise and reach (or even outperform) the quality of cascade approaches? At that time, the huge gap between the two paradigms portended a long way to go to reach performance parity. However, only a few months later, the yearly evaluation campaign organized by the IWSLT (Ansari et al., 2020) was won by a direct solution for the first time. Three years later, at the time of writing this thesis, the joint efforts of the growing research community, which covered many aspects (as described in §3 and §4), led to substantial performance parity between the two paradigms. In this endeavor, our contributions can be summarized as follows:

- We carried out an in-depth study on the best methods to transfer knowledge from an MT model into a direct ST system with knowledge distillation, highlighting not only the benefits but also its limitations, for which we provided an easy yet effective solution.
- We proposed a compression mechanism that leverages the prediction of a CTC module and dynamically reduces the length of the input sequence in the encoder of ST systems, improving both translation quality and computational efficiency.

- Featuring the CTC-compression module, we introduced Speechformer, the first architecture for direct ST that, enabled by an attention implementation with reduced computational complexity, avoids any fixed compression of the audio input, respecting the variability of the amount of information in speech signals and bringing significant quality gains.

- We demonstrated that we can obtain high translation quality even without ASR pre-training of the encoder and that a simple data filtering procedure significantly improves the quality of the resulting model.

- We increased the robustness of direct ST models with regard to automatic segmentation of the audio by fine-tuning them on resegmented training corpora and by providing the previous audio segment as contextual information;

- We proposed a new hybrid segmentation method that limits the quality degradation with respect to optimal segmentation based on the transcripts (unknown at inference time).

In the present condition of substantial parity in terms of translation quality between cascade and direct solutions, we believe that the preference for one paradigm may depend on other factors, such as the computational efficiency (hence the latency), the simplicity in training and using a system, its hardware requirements, data availability, or other peculiarities important for a specific domain. Accordingly, we speculate that future works will focus on the transfer of knowledge from foundation models (Bommasani et al., 2021) while keeping under control computational costs, a topic on which the first works recently started to appear (Le et al., 2021; Zhao et al., 2022). We also posit that computational efficiency will be a key factor for the widespread adoption of direct ST in production, as its advantage in terms of latency over the cascade paradigm makes it the natural candidate for streaming (or simultaneous) use cases.

The huge improvements in terms of translation quality allowed for pinpointed analyses of the capabilities of direct ST models. As such, an important part of the PhD has been devoted to the investigation of their behavior regarding two important aspects: gender bias (§5) and NEs (§6). In the first case, the goal was to ensure the fairness of automatic systems and equal opportunities for different groups of users in benefiting from them. As such, the work has been motivated by ethical principles that should always guide the development of technical solutions, and by the deep belief in the importance of raising the awareness of the limitations, and even potential harms, of automatically-generated text in contemporary society. Our contributions on the topic (see §5) include:

- The exploration of different solutions to control the grammatical gender of words referred to the speaker, investigating for the first time the case in which the speakers' gender conflicts with their vocal characteristics.
- The unveiling of the exacerbation of gender bias caused by a BPE segmentation of the target text in comparison with a character-based segmentation, and the proposal of a solution that goes beyond the trade-off between translation quality (BPE) and gender accuracy (char).
- A fine-grade evaluation of gender bias in ST showing that ST models are nearly perfect in handling gender agreement and the most biased part of speech is nouns.
- The investigation of the increase in gender bias caused by distilling knowledge from MT and how to solve the issue.

We think that current works have clearly shown that gender bias is a problem that is not only caused by biased corpora. Rather, algorithmic bias plays an important role in exacerbating representational differences. As such, also stimulated by the above contributions, we expect more investigation into this line of research in the future. In addition, while we mostly focused on remediation strategies involving training new models, an appealing alternative is the definition of inference-time solutions applicable to any (also existing) model.

At last, our second study on peculiar aspects (NEs, specifically) has been driven by the practical needs of professional interpreters in the context of the project Smarter Interpreting[1] financed by CDTI Neotec funds. Indeed, these users see in ST technology an opportunity to "augment" their capabilities and the quality of their work. Their needs, though, require dedicated research endeavors to highlight the potential of automatic systems on the specific aspects of interest, with the proposal of tailored solutions for the augmented ST scenario, neglected so far. The results of our activities on the topic (see §6) showed that:

- Cascade and direct ST systems behave similarly when it comes to NEs and they are particularly weak on person names, as demonstrated on our newly-created benchmark (NEurRoparl-ST).
- The most complex person names for ST systems are those with low frequency in the training data and those associated with languages not included in the source side of the training set, and multilingual models that jointly predict the transcript and the translation (giving more weight to the transcription) are more accurate in handling person names.
- In cases in which a dictionary of entities likely to appear in a given domain is available, the accuracy of NEs (especially of person names) can be significantly improved by means of additional modules that first

[1] `https://smarter-interpreting.eu/`

recognize which of them are present and then inject the corresponding translations as suggestions while generating the output.

- Models that jointly perform ST and NER outperform a pipeline of ST and NER systems while keeping the computational cost as low as that of a single ST model.

Our models have been presented in two demos, carried out in April and December 2022, in which our joint ST and NER systems have been integrated into a new CAI tool that displays the translated NEs and domain-specific terminology in real time to the interpreter. The dissemination of the work included its presentation at international conferences on interpreting[2], where it was introduced as 4th generation CAI system. The interest shown by the interpreting community in the solution persuades us to posit that more work on the topic will spread in the future with production tools including similar solutions in the mid-term. Moreover, as in many use cases additional contextual information is available at inference time, we believe more works on this line will appear, with the goal of integrating external knowledge without requiring excessive computational costs for their application to simultaneous settings.

7.1 Limitations and Future Directions

Direct ST Quality and Efficiency One limitation of our solutions lies in the scarce adoption of some of our methods, caused by different factors. First, word-level KD has not been widely used, as sequence-level KD is preferred due to its simplicity and similar efficacy. As such, only this latter method is included in the training procedure of recent ST systems (Anastasopoulos et al., 2022). Second, the Speechformer has been outperformed by the

[2] https://ctn.hkbu.edu.hk/interpreting_conf2022/

Conformer architecture, which has become the state of the art for ST. However, the same idea and concept of the Speechformer architecture can be applied to the Conformer. Although this integration did not prove effective in our experiments in §3.6, our recent discovery of bugs present in all the open-source implementations of the Conformer architecture related to padding management (Papi et al., 2023) may explain the reason of these negative results: as the Speechformer blocks do not compress the sequences, the amount of padding is higher and the negative impact of bugs related to padding management is likely to increase accordingly. In light of this, more investigations on the potential integration of the Speechformer into the Conformer architecture are needed. We take the opportunity to remark that, instead, the CTC compression is currently adopted in many works (sometimes with different naming, e.g. *shrinking*, or with minor modifications, e.g. ignoring vectors corresponding to the blank symbol) thanks to its benefits both in terms of quality and efficiency. In the future, we expect more and more solutions based on alternatives to the self-attention with reduced computational complexity, such as (Poli et al., 2023), to be the basis of models similar to the Speechformer, as well as more works on methods to compress the sequence length to improve the efficiency of ST models without penalizing quality.

Audio Segmentation Similarly, the introduction of SHAS has limited the adoption of our methods for audio segmentation. However, our methods and SHAS have mainly been tested in the TED domain, which typically involves a single speaker, minimal background noise, and ideal audio conditions. Additionally, the evaluation has primarily been conducted on English speech, and our findings have highlighted the need for different segmentations for different languages, as discussed in §4.4.4. In light of these limitations, future studies should evaluate the robustness of these methods to background noise and investigate its impact on the final task performance.

Gender Bias Our solutions to control the grammatical gender of words referred to the speaker depend on dedicated training procedures, which have inherent limitations. First, they require parallel audio-text data labeled with the speakers' gender, which are scarcely available and expensive to collect. Second, the training procedures have high resource requirements, as they involve processing audio data. Future work should overcome these limitations by proposing inference-time solutions that do not require dedicated training or data and can be applied to any existing model. In addition, our findings in terms of the effectiveness of debiasing techniques should be confirmed with newer architectures (e.g., Conformer), and, possibly, on a wider range of languages and domains. Indeed, our experiments leveraged MuST-SHE, a benchmark built on TED talks for three Latin language pairs (en→es,fr,it), which has only been partially adopted by the community, likely due to the limited number of languages covered.

Augmented Speech Translation Also in this case, our findings should be confirmed on a wider set of language pairs, as we were able to annotate with NEs only the en→es,fr,it sections of the Europarl-ST test set. We hope that our pioneering work on the topic will inspire other researchers to build more benchmarks on other domains and languages. Regarding our work on recognizing and injecting entities likely to appear in a given domain, two main weaknesses of our solution require future improvements for their deployment in real/production environments. First, the NE detection module has a linear complexity with the number of entities to check, therefore having a high computational cost in presence of large dictionaries of entities. Second, our injection method proved particularly helpful for person names, where the source and target representation is close, but should be improved to bring more benefits also on other categories with very different source and target text. However, given the importance of NEs for the reliability of the translation, we believe that this topic will receive more and more attention in the future, speeding up the process of finding new and better solutions.

Bibliography

Ruchit R. Agrawal, Marco Turchi, and Matteo Negri. Contextual Handling in Neural Machine Translation: Look Behind, Ahead and on Both Sides. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, pages 11–20, Alacant, Spain, May 2018.

Benyamin Ahmadnia, Bonnie J. Dorr, and Parisa Kordjamshidi. Knowledge Graphs Effectiveness in Neural Machine Translation Improvement. *Computer Science*, 21(3), September 2020. doi: 10.7494/csci.2020.21.3.3701. URL `https://journals.agh.edu.pl/csci/article/view/3701`.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Transla-*

tion, pages 1–88, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.1`.

Belen Alastruey, Gerard I. Gállego, and Marta Ruiz Costa-jussà. Efficient transformer for direct speech translation. *ArXiv*, abs/2107.03069, 2021.

Belen Alastruey, Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. On the locality of attention in direct speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 402–412, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-srw.32. URL `https://aclanthology.org/2022.acl-srw.32`.

Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno. Bringing contextual information to google speech recognition. In *Proceedings of Interspeech 2015*, pages 468–472, 2015. doi: 10.21437/Interspeech.2015-177.

Bashar Alhafni, Nizar Habash, and Houda Bouamor. Gender-Aware Reinflection using Linguistically Enhanced Neural Models. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.gebnlp-1.12`.

Diego Alves, Askars Salimbajevs, and Mārcis Pinnis. Data augmentation for pipeline-based speech translation. In *9th International Conference on Human Language Technologies - the Baltic Perspective (Baltic HLT 2020)*, Kaunas, Lithuania, September 2020. URL `https://hal.inria.fr/hal-02907053`.

Chantal Amrhein and Rico Sennrich. Identifying weaknesses in machine translation metrics through minimum bayes risk decoding: A case study for comet. *ArXiv*, abs/2202.05148, 2022.

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. Association for Computational Linguistics. 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*, abs/2208.07657, 2022.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, Washington, 2020.

Duygu Ataman and Marcello Federico. An Evaluation of Two Vocabulary Reduction Methods for Neural Machine Translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 97–110, Boston, MA, March 2018. Association for Machine Translation in the Americas. URL `https://www.aclweb.org/anthology/W18-1810`.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English. *The Prague Bulletin of Mathematical*

Linguistics, 108:331–342, 2017. URL http://ufal.mff.cuni.cz/pbml/108/art-ataman-negri-turchi-federico.pdf.

Duygu Ataman, Orhan Firat, Mattia A. Di Gangi, Marcello Federico, and Alexandra Birch. On the Importance of Word Boundaries in Character-level Neural Machine Translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 187–193, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5619. URL https://www.aclweb.org/anthology/D19-5619.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain Adaptation via Pseudo In-Domain Data Selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D11-1033.

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

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, Online, December 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

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 2019a.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. On Using SpecAugment for End-to-End Speech Translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, China, November 2019b.

Parnia Bahar, Patrick Wilken, Tamer Alkhouli, Andreas Guta, Pavel Golik, Evgeny Matusov, and Christian Herold. Start-Before-End and End-to-End: Neural Speech Translation by AppTek and RWTH Aachen University. In *Proceedings of 17th International Workshop on Spoken Language Translation (IWSLT)*, Virtual, July 2020.

Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. Tight Integrated End-to-End Training for Cascaded Speech Translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 950–957, 2021. doi: 10.1109/SLT48900.2021.9383462.

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*, May 2015.

Jeong-Uk Bang, Min-Kyu Lee, Seung Yun, and Sang-Hun Kim. Improving end-to-end speech translation model with bert-based contextual information. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6227–6231, 2022. doi: 10.1109/ICASSP43922.2022.9746117.

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. 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`.

Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. The Problem With Bias: Allocative Versus Representational Harms in Machine Learning. In *Proceedings of the 9th Annual Conference of the Special Interest Group for Computing, Information and Society (SIGCIS)*, Philadelphia, Pennsylvania, October 2017.

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 Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1118.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. On the Linguistic Representational Power of Neural Machine Translation Models. *Computational Linguistics*, 46(1):1–52, 2020. doi: 10.1162/coli_a_00367. URL `https://doi.org/10.1162/coli_a_00367`.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL `https://doi.org/10.1145/3442188.3445922`.

Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. The problem of learning long-term dependencies in recurrent networks. In *IEEE International Conference on Neural Networks*, pages 1183–1188 vol.3, 1993. doi: 10.1109/ICNN.1993.298725.

Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5 2:157–66, 1994.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. Gender in Danger? Evaluating Speech Translation Technology on the MuST-SHE Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-main.619`.

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 International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2873–2887, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.224. URL `https://aclanthology.org/2021.acl-long.224`.

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.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-End Automatic Speech Translation of Audiobooks. In *Proceedings of ICASSP 2018*, pages 6224–6228, Calgary, Alberta, Canada, April 2018.

Nicola Bertoldi, Roldano Cattoni, Mauro Cettolo, Amin Farajian, Marcello Federico, Davide Caroselli, Luca Mastrostefano, Andrea Rossi, Marco Trombetti, Ulrich Germann, and David Madl. MMT: New Open Source MT for the Translation Industry. In *Proceedings of the 20th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 86–91, Prague, Czech Republic, May 2017.

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.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (Technology) is Power: A Critical Survey of “Bias” in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.485. URL `https://www.aclweb.org/anthology/2020.acl-main.485`.

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. Association for Computational Linguistics. doi: 10.3115/v1/W14-3302. URL https://aclanthology.org/W14-3302.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021. URL https://arxiv.org/abs/2108.07258.

Léon Bottou. *On-line Learning and Stochastic Approximations*, page 9–42. Publications of the Newton Institute. Cambridge University Press, 1999. doi: 10.1017/CBO9780511569920.003.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. Routledge, 1984. doi: https://doi.org/10.1201/9781315139470.

Antoine Bruguier, Fuchun Peng, and Françoise Beaufays. Learning Personalized Pronunciations for Contact Name Recognition. In *Interspeech 2016*, pages 3096–3100, 2016. doi: 10.21437/Interspeech.2016-537. URL http://dx.doi.org/10.21437/Interspeech.2016-537.

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.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. DeepPavlov: Open-Source Library for Dialogue Systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4021. URL https://www.aclweb.org/anthology/P18-4021.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the Role of Bleu in Machine Translation Research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E06-1032.

Víctor Campos, Brendan Jou, Xavier Giro-i Nieto, Jordi Torres, and Shih-Fu Chang. Skip rnn: Learning to skip state updates in recurrent neural networks. In *Proceedings of 6th International Conference on Learning Representations (ICLR)*, 2018.

Yang T. Cao and Hal Daumé III. Toward Gender-Inclusive Coreference Resolution. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.418. URL https://www.aclweb.org/anthology/2020.acl-main.418.

Antoine Caubrière, Sophie Rosset, Yannick Estève, Antoine Laurent, and Emmanuel Morin. Where are we in Named Entity Recognition from Speech? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4514–4520, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.556.

Augustin-Louis Cauchy. Methode generale pour la resolution des systemes d'equations simultanees. *C.R. Acad. Sci. Paris*, 25:536–538, 1847.

Feng-Ju Chang, Jing Liu, Martin Radfar, Athanasios Mouchtaris, Maurizio Omologo, Ariya Rastrow, and Siegfried Kunzmann. Context-aware transformer transducer for speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 503–510, 2021. doi: 10.1109/ASRU51503.2021.9687895.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. Guiding Neural Machine Translation Decoding with External Knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4716. URL `https://aclanthology.org/W17-4716`.

Boli Chen, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Meishan Zhang, and Fei Huang. Aishell-ner: Named entity recognition from chinese speech. In *Proceedings of ICASSP 2022*, pages 8352–8356, 2022. doi: 10.1109/ICASSP43922.2022.9746955.

Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L. Seltzer, and Christian Fuegen. End-to-end contextual speech recognition using class language models and a token passing decoder. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6186–6190, 2019. doi: 10.1109/ICASSP.2019.8683573.

Agnieszka Chmiel, Agnieszka Szarkowska, Danijel Korzinek, Agnieszka Lijewska, Łukasz Dutka, Łukasz Brocki, and Krzysztof 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.

Eunah Cho, Jan Niehues, and Alex Waibel. NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation. In *Proceeding of Interspeech 2017*, pages 2645–2649, Stockholm, Sweden, August 2017. doi: 10.21437/Interspeech.2017-1320.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://aclanthology.org/D14-1179`.

Won Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. On Measuring Gender Bias in Translation of Gender-neutral Pronouns. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 173–181, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3824. URL `https://www.aclweb.org/anthology/W19-3824`.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021*, Virtual Event, Austria, May 2021. URL `https://openreview.net/forum?id=Ua6zuk0WRH`.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

John W. Chotlos. A statistical and comparative analysis of individual written language samples. *Psychological Monographs*, 56(2):75, 1944.

Shun-Po Chuang, Tzu-Wei Sung, Alexander H. Liu, and Hung-yi Lee. Worse

WER, but Better BLEU? Leveraging Word Embedding as Intermediate in Multitask End-to-End Speech Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5998–6003, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.533. URL `https://aclanthology.org/2020.acl-main.533`.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1160. URL `https://www.aclweb.org/anthology/P16-1160`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537, nov 2011. ISSN 1532-4435.

Bernard Comrie. Grammatical gender systems: a linguist's assessment. *Journal of Psycholinguistic research*, 28(5):457–466, 1999.

Greville G. Corbett. *Gender*. Cambridge University Press, Cambridge, UK, 1991.

Greville G. Corbett. *Agreement*. Cambridge University Press, 2006.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2058. URL https://aclanthology.org/P16-2058.

Marta R. Costa-jussà, Carlos Escolano, Christine Basta, Javier Ferrando, Roser Batlle, and Ksenia Kharitonova. Gender Bias in Multilingual Neural Machine Translation: The Architecture Matters. *arXiv preprint arXiv:2012.13176*, 2020.

Marta R. Costa-jussà, Christine Basta, and Gerard I. Gállego. Evaluating gender bias in speech translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2141–2147, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.230.

Marta R. Costa-jussà. An Analysis of Gender Bias Studies in Natural Language Processing. *Nature Machine Intelligence*, 1:495–496, 2019. URL https://www.nature.com/articles/s42256-019-0105-5.

Michael A. Covington and Joe D. McFall. Cutting the Gordian knot: The moving-average type–token ratio (MATTR). *Journal of quantitative linguistics*, 17(2):94–100, 2010.

Kate Crawford. The Trouble with Bias. In *Conference on Neural Information Processing Systems (NIPS) – Keynote*, Long Beach, California, December 2017. URL https://www.youtube.com/watch?v=fMym_BKWQzk.

Mathias Creutz and Krista Lagus. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. *International Symposium on Computer and Information Sciences*, 2005.

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. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL `https://aclanthology.org/P19-1285`.

Arjun Das, Debasis Ganguly, and Utpal Garain. Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 16(3), January 2017. ISSN 2375-4699. doi: 10.1145/3015467. URL `https://doi.org/10.1145/3015467`.

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.

Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4): 357–366, 1980. doi: 10.1109/TASSP.1980.1163420.

Bart Desmet, Mieke Vandierendonck, and Bart Defrancq. Simultaneous interpretation of numbers and the impact of technological support. In Claudio Fantinuoli, editor, *Interpreting and technology*, Translation and Multilingual Natural Language Processing, pages 13–27. Language Science Press, 2018. ISBN 9783961101627. URL `http://dx.doi.org/10.5281/zenodo.1493281`.

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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, page 2012–2017, Minneapolis, Minnesota, June 2019a.

Mattia A. Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. Enhancing transformer for end-to-end speech-to-text translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 21–31, Dublin, Ireland, August 2019b. European Association for Machine Translation. URL `https://aclanthology.org/W19-6603`.

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

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. One-To-Many Multilingual End-to-end Speech Translation. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 585–592, Sentosa, Singapore, December 2019d.

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 2020a. Association for Machine Translation in the Americas. URL `https://www.aclweb.org/anthology/2020.amta-research.13`.

Mattia A. Di Gangi, Viet Nguyen, Matteo Negri, and Marco Turchi. Instance-based Model Adaptation for Direct Speech Translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7914–7918, Barcelona, Spain, May 2020b.

Jorge Diaz-Cintas and Aline Remael. *Audiovisual Translation: Subtitling*. Translation practices explained. Routledge, 2007.

Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. Training Neural Machine Translation to Apply Terminology Constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1294. URL `https://www.aclweb.org/anthology/P19-1294`.

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.

Duane K. Dougal and Deryle Lonsdale. Improving NMT Quality Using Terminology Injection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4820–4827, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.593`.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine*

Learning Research, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Stephanie Díaz-Galaz, Presentacion Padilla, and María Teresa Bajo. The role of advance preparation in simultaneous interpreting: A comparison of professional interpreters and interpreting students. *Interpreting*, 17 (1):1–25, 2015. ISSN 1384-6647. doi: https://doi.org/10.1075/intp.17.1.01dia. URL https://www.jbe-platform.com/content/journals/10.1075/intp.17.1.01dia.

Mostafa Elaraby, Ahmed Y. Tawfik, Mahmoud Khaled, Hany Hassan, and Aly Osama. Gender Aware Spoken Language Translation Applied to English-Arabic. In *Proceedings of the 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–6, Algiers, Algeria, 2018.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

Claudio Fantinuoli. *Chapter 7: Computer-assisted Interpreting: Challenges and Future Perspectives*, pages 153–174. Brill, Leiden, The Netherlands, 2017. ISBN 9789004351790. doi: https://doi.org/10.1163/9789004351790_009.

Claudio Fantinuoli and Bianca Prandi. Towards the evaluation of automatic simultaneous speech translation from a communicative perspective. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 245–254, Bangkok, Thailand (online), August 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. In *Proceedings of Translator and Computer 43 Conference*, 2022.

Christian Fügen. *A System for Simultaneous Translation of Lectures and Speeches*. PhD thesis, Universität Karlsruhe, Karlsruhe, 2009.

Ryo Fukuda, Katsuhito Sudoh, and Satoshi Nakamura. Speech Segmentation Optimization using Segmented Bilingual Speech Corpus for End-to-end Speech Translation. In *Proceedings of Interspeech 2022*, pages 121–125, 2022. doi: 10.21437/Interspeech.2022-11382.

Sadaoki Furui. Chapter 7 - speaker recognition in smart environments. In Hamid Aghajan, Ramón López-Cózar Delgado, and Juan Carlos Augusto, editors, *Human-Centric Interfaces for Ambient Intelligence*, pages 163–184. Academic Press, Oxford, 2010. ISBN 978-0-12-374708-2. doi: https://doi.org/10.1016/B978-0-12-374708-2.00007-3. URL `https://www.sciencedirect.com/science/article/pii/B9780123747082000073`.

Olivier Galibert, Jeremy Leixa, Gilles Adda, Khalid Choukri, and Guillaume Gravier. The ETAPE speech processing evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3995–3999, Reykjavik, Iceland, May 2014. European

Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1027_Paper.pdf.

Gerard I. Gállego, Ioannis Tsiamas, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. End-to-end speech translation with pre-trained models and adapters: UPC at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119, Bangkok, Thailand (online), August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwslt-1.11. URL https://aclanthology.org/2021.iwslt-1.11.

Mahault Garnerin, Solange Rossato, and Laurent Besacier. Gender Representation in French Broadcast Corpora and Its Impact on ASR Performance. In *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery*, AI4TV '19, page 3–9, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369176. doi: 10.1145/3347449.3357480. URL https://doi.org/10.1145/3347449.3357480.

Mahault Garnerin, Solange Rossato, and Laurent Besacier. Gender Representation in Open Source Speech Resources. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6599–6605, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.813.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673, Nov 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, and Emmanuel Morin. End-to-end named entity and semantic concept extraction from speech. In *Proceedings of 2018 IEEE SLT Workshop)*, pages 692–699, 2018. doi: 10.1109/SLT.2018.8639513.

Daniel Gile. Ad hoc knowledge acquisition in interpreting and translation. In *Basic Concepts and Models for Interpreter and Translator Training: Revised edition*, Benjamins Translation Library, chapter 6. John Benjamins Publishing Company, 2009.

GLAAD. Media Reference Guide - Transgender. 2007. URL `https://www.glaad.org/reference/transgender`.

Bruce Glymour and Jonathan Herington. Measuring the Biases that Matter: The Ethical and Casual Foundations for Measures of Fairness in Algorithms. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 269–278, 2019.

Anita Gojun and Alexander Fraser. Determining the placement of German verbs in English–to–German SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 726–735, Avignon, France, April 2012.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Aditya Gourav, Linda Liu, Ankur Gandhe, Yile Gu, Guitang Lan, Xiangyang Huang, Shashank Kalmane, Gautam Tiwari, Denis Filimonov, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko. Personalization strategies for end-to-end speech recognition systems. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Process-*

ing (ICASSP), pages 7348–7352, 2021. doi: 10.1109/ICASSP39728.2021.9413962.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/graves14.html`.

Alex Graves, Santiago Fernández, Faustino J. 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)*, pages 369–376, Pittsburgh, Pennsylvania, June 2006.

Zenzi M. Griffin and Kathryn Bock. Constraint, Word Frequency, and the Relationship between Lexical Processing Levels in Spoken Word Production. *Journal of Memory and Language*, 38(3):313–338, 1998. ISSN 0749-596X. doi: https://doi.org/10.1006/jmla.1997.2547. URL `https://www.sciencedirect.com/science/article/pii/S0749596X9792547X`.

Jiatao Gu and Xu Tan. Non-autoregressive sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 21–27, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-tutorials.4. URL `https://aclanthology.org/2022.acl-tutorials.4`.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1l8BtlCb`.

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 *Proceedings of Interspeech 2020*, pages 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

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.

Pascal M. Gygax, Daniel Elmiger, Sandrine Zufferey, Alan Garnham, Sabine Sczesny, Lisa von Stockhausen, Friederike Braun, and Jane Oakhill. A Language Index of Grammatical Gender Dimensions to Study the Impact of Grammatical Gender on the Way We Perceive Women and Men. *Frontiers in Psychology*, 10:1604, 2019. ISSN 1664-1078. doi: 10.3389/fpsyg.2019.01604. URL https://www.frontiersin.org/article/10.3389/fpsyg.2019.01604.

Nizar Habash, Houda Bouamor, and Christine Chung. Automatic Gender Identification and Reinflection in Arabic. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 155–165, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3822. URL https://www.aclweb.org/anthology/W19-3822.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.195. URL https://aclanthology.org/2021.findings-acl.195.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1108.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. Neural Machine Translation Decoding with Terminology Constraints. 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 506–512, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2081. URL https://www.aclweb.org/anthology/N18-2081.

William N. Havard, Jean-Pierre Chevrot, and Laurent Besacier. Catplayinginthesnow: Impact of Prior Segmentation on a Model of Visually Grounded Speech, 2020.

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.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. Dynamic Programming Encoding for Subword Segmentation in Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.275. URL https://www.aclweb.org/anthology/2020.acl-main.275.

Marlis Hellinger and Hadumond Bußman. *Gender across Languages.* John Benjamins Publishing, Amsterdam, The Netherlands, 2001.

Ulf Hermjakob, Kevin Knight, and Hal Daumé III. Name Translation in Statistical Machine Translation - Learning When to Transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P08-1045`.

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 *Proceedings of the Speech and Computer - 20th International Conference (SPECOM)*, pages 198–208, Leipzig, Germany, September 2018. ISBN 9783319995793. URL `http://dx.doi.org/10.1007/978-3-319-99579-3_21`.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. doi: 10.1109/MSP.2012.2205597.

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`.

Charles F. Hockett. *A Course in Modern Linguistics.* Macmillan, New York, New York, 1958.

Chris Hokamp and Qun Liu. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1535–1546, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1141. URL https://www.aclweb.org/anthology/P17-1141.

Laura Hollink, Astrid van Aggelen, Henri Beunders, Martijn Kleppe, Max Kemman, and Jacco van Ossenbruggen. Talk of Europe - The debates of the European Parliament as Linked Open Data, 2017.

Dirk Hovy and Shannon L. Spruit. The Social Impact of Natural Language Processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2096. URL https://www.aclweb.org/anthology/P16-2096.

Dirk Hovy, Federico Bianchi, and Tommaso Fornaciari. "You Sound Just Like Your Father" Commercial Machine Translation Systems Include Stylistic Biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1686–1690, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.154. URL https://www.aclweb.org/anthology/2020.acl-main.154.

Rongqing Huang, Ossama Abdel-hamid, Xinwei Li, and Gunnar Evermann. Class LM and Word Mapping for Contextual Biasing in End-to-End ASR. In *Proceedings of Interspeech 2020*, pages 4348–4351, 2020. doi: 10.21437/Interspeech.2020-1787.

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, December 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. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.34. URL `https://www.aclweb.org/anthology/2020.acl-demos.34`.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1872–1881, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.150. URL `https://aclanthology.org/2021.naacl-main.150`.

Sathish Indurthi, Houjeung Han, Nikhil Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. End-end speech-to-text translation with modality agnostic meta-learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7904–7908, 2020. doi: 10.1109/ICASSP40776.2020.9054759.

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, Joan A. Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Giménez. Adrià, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233, Barcelona, Spain, May 2020. URL `https://ieeexplore.ieee.org/document/9054626`.

Ray S. Jackendoff. *Semantic Structures.* Cambridge: MIT Press, 1990.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *ICML*, pages 4651–4664, Virtual Event, July 2021. PMLR. URL `http://proceedings.mlr.press/v139/jaegle21a.html`.

Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf. Contextual RNN-T for Open Domain ASR. In *Proceedings of Interspeech 2020*, pages 11–15, 2020. doi: 10.21437/Interspeech.2020-2986.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1001. URL `https://aclanthology.org/P15-1001`.

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.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL `https://www.aclweb.org/anthology/Q17-1024`.

Roderick Jones. Conference interpreting explained. *Interpreting*, 3(2): 201–203, 1998. ISSN 1384-6647. doi: https://doi.org/10.1075/intp.3.2.05mac. URL `https://www.jbe-platform.com/content/journals/10.1075/intp.3.2.05mac`.

Bing-Hwang Juang, S. Levinson, and M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.). *IEEE Transactions on Information Theory*, 32(2):307–309, 1986. doi: 10.1109/TIT.1986.1057145.

Namkyu Jung, Geonmin Kim, and Joon Son Chung. Spell my name: Keyword boosted speech recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6642–6646, 2022. doi: 10.1109/ICASSP43922.2022.9747714.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, Tatiana Likhomanenko, Gabriel Synnaeve, Armand Joulin, Abdelrahman Mohamed, and Emmanuel Dupoux.

Libri-light: A benchmark for asr with limited or no supervision. In *Proceedings of ICASSP 2020*, pages 7669–7673, 2020.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://aclanthology.org/D13-1176`.

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 2020. doi: 10.18653/v1/2020.iwslt-1.26.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, 2020.

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, mar 2017.

Yoon Kim and Alexander M. Rush. Sequence-Level Knowledge Distillation. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language*

Processing, pages 1317–1327, Austin, Texas, 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. Association for Computational Linguistics. doi: 10.18653/v1/D19-6503.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, San Diego, California, dec 2015.

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.

Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. Gender Coreference and Bias Evaluation at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 357–364, 2020.

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. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-3250`.

Philipp Koehn. Neural machine translation, 2017. URL `https://arxiv.org/abs/1709.07809`.

Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural*

Machine Translation, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL `https://www.aclweb.org/anthology/W17-3204`.

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`.

John F. Kolen and Stefan C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*, pages 237–243. 2001. doi: 10.1109/9780470544037.ch14.

Stefan C. Kremer and John F. Kolen. *Field Guide to Dynamical Recurrent Networks*. Wiley-IEEE Press, 1st edition, 2001. ISBN 0780353692.

Taku Kudo. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018.

Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. URL https://www.aclweb.org/anthology/P18-1007.

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: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012.

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *Ann. Math. Statist.*, 22(1):79–86, mar 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Paul Lamere, Philip Kwok, Evandro Gouv, Rita Singh, William Walker, and Peter Wolf. The cmu sphinx4 speech recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*, volume 1, pages 2–5, April 2003.

J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2529310.

Brian Larson. Gender as a variable in Natural-Language Processing: Ethical considerations. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 1–11, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1601. URL https://www.aclweb.org/anthology/W17-1601.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 817–824, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.103. URL `https://aclanthology.org/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`.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378, 2017. doi: 10.1162/tacl_a_00067. URL `https://aclanthology.org/Q17-1026`.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in Neural Machine Translation. In *Proceedings of NIPS 2018 Interpretability and Robustness for Audio, Speech and Language Workshop*, Montréal, Canada, December 2018.

Enora Lessinger. The Challenges of Translating Gender in UN texts. In Luise von Flotow and Hala Kamal, editors, *The Routledge Handbook of Translation, Feminism and Gender*. Routledge, New York, NY, USA, 2020.

Hector J. Levesque. On Our Best Behaviour. *Artif. Intell.*, 212(1):27–35, July 2014. ISSN 0004-3702. doi: 10.1016/j.artint.2014.03.007. URL `https://doi.org/10.1016/j.artint.2014.03.007`.

Sarah I. Levitan, Taniya Mishra, and Srinivas Bangalore. Automatic identification of gender from speech. In *In Proocedings of Speech Prosody 2016*, pages 84–88, Boston, Massachusetts, May-June 2016. doi: 10.21437/SpeechProsody.2016-18. URL `http://dx.doi.org/10.21437/SpeechProsody.2016-18`.

Haibo Li, Jing Zheng, Heng Ji, Qi Li, and Wen Wang. Name-aware Machine Translation. In *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://www.aclweb.org/anthology/P13-1059`.

Xintong Li, Lemao Liu, Rui Wang, Guoping Huang, and Max Meng. Regularized context gates on transformer for machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8555–8562, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.757. URL `https://aclanthology.org/2020.acl-main.757`.

Minhua Liu, Diane L. Schallert, and Patrick J. Carroll. Working memory and expertise in simultaneous interpreting. *Interpreting*, 6(1): 19–42, 2004. ISSN 1384-6647. doi: https://doi.org/10.1075/intp.6.1.04liu. URL `https://www.jbe-platform.com/content/journals/10.1075/intp.6.1.04liu`.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-End Speech Translation with Knowledge Distillation. In *Proceedings of Interspeech 2019*, pages 1128–1132, Graz, Austria, sep 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, 2020.

Arle Lommel. Augmented translation: A new approach to combining human and machine capabilities. In *Proc. of the 13th AMTA*, pages 5–12, Boston, MA, 2018. 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, 2019a. URL https://arxiv.org/abs/1906.02762.

Yu Lu, Jiajun Zhang, and Chengqing Zong. Exploiting knowledge graph in neural machine translation. In Jiajun Chen and Jiajun Zhang, editors, *Machine Translation*, pages 27–38, Singapore, 2019b. Springer Singapore. ISBN 978-981-13-3083-4.

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. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166.

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 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.aacl-main.58.

Kate MacKrill, Connor Silvester, James W Pennebaker, and Keith J Petrie. What makes an idea worth spreading? language markers of popularity in ted talks by academics and other speakers. *Journal of the Association for Information Science and Technology*, 2021.

Miquel Martínez De Morentin Cardoner. *Context-Aware End-to-End Speech Translation.* PhD thesis, UPC, Facultat d'Informàtica de Barcelona, Departament de Teoria del Senyal i Comunicacions, June 2022. URL `http://hdl.handle.net/2117/375119`.

Evgeny Matusov, Arne Mauser, and Hermann Ney. Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT) 2006*, pages 158–165, Kyoto, Japan, November 2006.

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. Association for Computational Linguistics. doi: 10.18653/v1/W19-5209. URL `https://www.aclweb.org/anthology/W19-5209`.

Michael Mccloskey and Neil J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 24:104–169, 1989.

Sylvain Meignier and Teva Merlin. LIUM SpkDiarization: An Open Source Toolkit For Diarization. In *Proceedings of the CMU SPUD Workshop*, Dallas, Texas, March 2010.

Tom M. Mitchell. The need for biases in learning generalizations. In Jude W. Shavlik and Thomas G. Dietterich, editors, *Readings in Machine Learning*, pages 184–191. Morgan Kauffman, 1980.

Maciej Modrzejewski, Miriam Exel, Bianka Buschbeck, Thanh-Le Ha, and Alexander Waibel. Incorporating external annotation to improve named entity translation in NMT. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 45–51, Lisboa, Portugal, November 2020. European Association for Machine Translation. URL `https://aclanthology.org/2020.eamt-1.6`.

Amit Moryossef, Roee Aharoni, and Yoav Goldberg. Filling Gender & Number Gaps in Neural Machine Translation with Black-Box Context Injection. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 49–54, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3807. URL `https://www.aclweb.org/anthology/W19-3807`.

Diego Moussallem, Axel-Cyrille Ngonga Ngomo, Paul Buitelaar, and Mihael Arcan. Utilizing knowledge graphs for neural machine translation augmentation. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 139–146, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370080. doi: 10.1145/3360901.3364423. URL `https://doi.org/10.1145/3360901.3364423`.

Rui Na, Junfeng Hou, Wu Guo, Yan Song, and Lirong Dai. Learning adaptive downsampling encoding for online end-to-end speech recognition. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 850–854, 2019. doi: 10.1109/APSIPAASC47483.2019.9023043.

Jiří Navrátil, Karthik Visweswariah, and Ananthakrishnan Ramanathan. A comparison of syntactic reordering methods for English-German machine translation. In *Proceedings of COLING 2012*, pages 2043–2058, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1125.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. XNMT: The eXtensible Neural Machine Translation Toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pages 185–192, Boston, MA, March 2018. URL https://www.aclweb.org/anthology/W18-1818.

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 2020.

Jan Niehues. Continuous learning in neural machine translation using bilingual dictionaries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 830–840, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.70. URL https://aclanthology.org/2021.eacl-main.70.

Jan Niehues, Rolando Cattoni, Sebastian Stüker, Mauro Cettolo, Marco Turchi, and Marcello Federico. The IWSLT 2018 evaluation campaign. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 2–6, Brussels, October 29-30 2018. International Conference on Spoken Language Translation. URL https://aclanthology.org/2018.iwslt-1.1.

Jan Niehues, Roldano Cattoni, Sebastian Stucker, 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 16th International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, nov 2019. URL `https://doi.org/10.5281/zenodo.3525578`.

Chikashi Nobata, Nigel Collier, and Jun'ichi Tsujii. Comparison between tagged corpora for the named entity task. In *The Workshop on Comparing Corpora*, pages 20–27, Hong Kong, China, October 2000. Association for Computational Linguistics. doi: 10.3115/1117729.1117733. URL `https://aclanthology.org/W00-0904`.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. Dynamic grammars with lookahead composition for WFST-based speech recognition. In *Proceedings of Interspeech 2012*, pages 1079–1082, 2012. doi: 10.21437/Interspeech.2012-327.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Optimizing Segmentation Strategies for Simultaneous Speech Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2090.

Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice-hall Englewood Cliffs, second edition, 1999.

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 Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June

2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL `https://www.aclweb.org/anthology/N19-4009`.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, Queensland, Australia, apr 2015.

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 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.autosimtrans-1.2. URL `https://aclanthology.org/2022.autosimtrans-1.2`.

Sara Papi, Marco Gaido, Andrea Pilzer, and Matteo Negri. Reproducibility is Nothing without Correctness: The Importance of Testing Code in NLP, 2023.

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 Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://www.aclweb.org/anthology/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 *Proceedings of Interspeech 2019*, pages 2613–2617, Graz, Austria, September 2019. doi:

10.21437/Interspeech.2019-2680. URL http://dx.doi.org/10.21437/Interspeech.2019-2680.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

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. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.24. URL https://aclanthology.org/2022.iwslt-1.24.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models, 2023.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(64)90137-5. URL https://www.sciencedirect.com/science/article/pii/0041555364901375.

Dejan Porjazovski, Juho Leinonen, and Mikko Kurimo. Attention-Based End-to-End Named Entity Recognition from Speech. In Kamil Ekštein, František Pártl, and Miloslav Konopík, editors, *Text, Speech, and Dialogue*, pages 469–480, Cham, 2021. Springer International Publishing. ISBN 978-3-030-83527-9.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL `https://www.aclweb.org/anthology/W18-6319`.

Matt Post, Shuoyang Ding, Marianna Martindale, and Winston Wu. An exploration of placeholding in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 182–192, Dublin, Ireland, August 2019. European Association for Machine Translation. URL `https://aclanthology.org/W19-6618`.

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. Association for Computational Linguistics. doi: 10.18653/v1/2020.iwslt-1.9. URL `https://www.aclweb.org/anthology/2020.iwslt-1.9`.

Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur. A Time-Restricted Self-Attention Layer for ASR. In *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878, 2018.

Bianca Prandi. An exploratory study on CAI tools in simultaneous interpreting: Theoretical framework and stimulus validation, November 2018. URL `https://doi.org/10.5281/zenodo.1493293`.

Marcelo O. R. Prates, Pedro H. C. Avelar, and Luís C. Lamb. Assessing Gender Bias in Machine Translation: a Case Study with Google Translate. *Neural Computing and Applications*, 32(10):6363–6381, 2020.

Golan Pundak, Tara N. Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. Deep context: End-to-end contextual speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 418–425, 2018. doi: 10.1109/SLT.2018.8639034.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Hema Raghavan and James Allan. Matching Inconsistently Spelled Names in Automatic Speech Recognizer Output for Information Retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 451–458, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL https://aclanthology.org/H05-1057.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

Lance Ramshaw and Mitch Marcus. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*, 1995. URL https://www.aclweb.org/anthology/W95-0107.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *Proceedings of 4th International Conference on Learning Representations ICLR*, San Juan, Puerto Rico, May 2-4 2016. URL https://arxiv.org/abs/1511.06732.

Vijay Ravi, Yile Gu, Ankur Gandhe, Ariya Rastrow, Linda Liu, Denis Filimonov, Scott Novotney, and Ivan Bulyko. Improving accuracy of rare words for rnn-transducer through unigram shallow fusion, 2020. URL `https://arxiv.org/abs/2012.00133`.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL `https://aclanthology.org/2020.emnlp-main.213`.

Christina Richards, Walter P. Bouman, Leighton Seal, Meg John Barker, Timo O. Nieder, and Guy T'Sjoen. Non-binary or Genderqueer Genders. *International Review of Psychiatry*, 28(1):95–102, 2016.

Nicholas Roberts, Davis Liang, Graham Neubig, and Zachary C. Lipton. Decoding and Diversity in Machine Translation. In *Proceedings of the Resistance AI Workshop at 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, February 2020.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. URL `https://arxiv.org/abs/1609.04747`.

Nicholas Ruiz and Marcello Federico. Assessing the impact of speech recognition errors on machine translation quality. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*, pages 261–274, Vancouver, Canada, October 22-26 2014. Association for Machine Translation in the Americas. URL `https://aclanthology.org/2014.amta-researchers.20`.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. In *Proceedings of Interspeech 2015*, Dresden, Germany, September 2015.

Elizabeth Salesky and Alan W. Black. Phone Features Improve Speech Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2388–2397, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.217.

Elizabeth Salesky, Matthias Sperber, and Alan W. Black. Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1179.

Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. The Multilingual TEDx Corpus for Speech Recognition and Translation. In *Proceedings of Interspeech 2021*, pages 3655–3659, 2021. doi: 10.21437/Interspeech.2021-11.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: A Large-scale Dataset For Multimodal Language Understanding. In *Proceedings of Visually Grounded Interaction and Language (ViGIL)*, Montréal, Canada, dec 2018. Neural Information Processing Society (NeurIPS). URL https://hal.archives-ouvertes.fr/hal-02431947.

Danielle Saunders and Bill Byrne. Reducing Gender Bias in Neural Machine Translation as a Domain Adaptation Problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.690. URL https://www.aclweb.org/anthology/2020.acl-main.690.

Danielle Saunders, Rosie Sallis, and Bill Byrne. Neural Machine Translation Doesn't Translate Gender Coreference Right Unless You Make It. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 35–43, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.gebnlp-1.4.

Paul Schachter and Timothy Shopen. Parts-of-speech systems. *Language Typology and Syntactic Description. Vol. 1: Clause Structure*, pages 1–60, 2007.

Florian Schiel. Automatic Phonetic Transcription of Non-Prompted Speech. In John J. Ohala, editor, *Proceedings of the XIVth International Congress of Phonetic Sciences: ICPhS 99*, pages 607 –610, San Francisco, California, August 1999. URL http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-13682-6.

Kristen Schilt and Lauren Westbrook. "Gender Normals," Transgender People, and the Social Maintenance of Heterosexuality. *Gender & Society*, 23(4), 2009. URL https://doi.org/10.1177/0891243209340034. 440-464.

Juergen Schmidhuber. Self-delimiting neural networks, 2012. URL https://arxiv.org/abs/1210.0118.

William A. Scott. Reliability of content analysis: The case of nominal scale coding. *Pubulic Opinion Quarterly*, 19:321–325, 1955.

Terrence J. Sejnowski. *The Deep Learning Revolution.* MIT Press, Cambridge, MA, 2018. ISBN 978-0-262-03803-4.

Rico Sennrich. How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E17-2060`.

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. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

Deven S. Shah, Hansen A. Schwartz, and Dirk Hovy. Predictive Biases in Natural Language Processing Models: A Conceptual Framework and Overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-main.468`.

Mark Sinclair, Peter Bell, Alexandra Birch, and Fergus Mcinnes. A semi-Markov model for speech segmentation with an utterance-break prior. In *Proceedings of Interpseech 2014*, pages 2351–2355, Singapore, September 2014.

Fabian H. Sinz, Xaq Pitkow, Jacob Reimer, Matthias Bethge, and Andreas S. Tolias. Engineering a Less Artificial Intelligence. *Neuron*, 103(6):967–979, 2019. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2019.08.034. URL `https://www.sciencedirect.com/science/article/pii/S0896627319307408`.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-2006. URL `https://www.aclweb.org/anthology/E14-2006`.

Matthew Snover, Bonnie Dorr, Richard 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,*, pages 223–231, Cambridge, August 2006. Association for Machine Translation in the Americas.

David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V Le. Searching for efficient transformers for language modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6010–6022. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/2f3c6a4cd8af177f6456e7e51a916ff3-Paper.pdf`.

Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3, 1999.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua

Luo, Xiangyu Duan, and Min Zhang. Alignment-Enhanced Transformer for Constraining NMT with Pre-Specified Translations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8886–8893, Apr. 2020. doi: 10.1609/aaai.v34i05.6418. URL https://ojs.aaai.org/index.php/AAAI/article/view/6418.

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. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.661. URL https://aclanthology.org/2020.acl-main.661.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. Self-Attentional Acoustic Models. In *Proceedings of Interspeech 2018*, pages 3723–3727, 2018. doi: 10.21437/Interspeech.2018-1910. URL http://dx.doi.org/10.21437/Interspeech.2018-1910.

Matthias Sperber, Hendra Setiawan, Christian Gollan, Udhyakumar Nallasamy, and Matthias Paulik. Consistent Transcription and Translation of Speech. *Transactions of the Association for Computational Linguistics*, 8:695–709, 2020. doi: 10.1162/tacl_a_00340. URL https://www.aclweb.org/anthology/2020.tacl-1.45.

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.

Artūrs Stafanovičs, Mārcis Pinnis, and Toms Bergmanis. Mitigating Gender Bias in Machine Translation with Target Gender Annotations. In *Proceedings of the Fifth Conference on Machine Translation*, pages 629–638, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.wmt-1.73.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. Evaluating Gender Bias in Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1164. URL https://www.aclweb.org/anthology/P19-1164.

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

Craig Stewart, Nikolai Vogler, Junjie Hu, Jordan Boyd-Graber, and Graham Neubig. Automatic Estimation of Simultaneous Interpreter Performance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 662–666, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2105. URL https://aclanthology.org/P18-2105.

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.

Susan Stryker. Transgender history, homonormativity, and disciplinarity. *Radical History Review*, 2008(100):145–157, 2008.

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908. ISSN 00063444. URL http://www.jstor.org/stable/2331554.

Atiwong Suchato, Proadpran Punyabukkana, Patanan Ariyakornwijit, and Teerat Namchaisawatwong. Automatic speech recognition of Thai person names from dynamic name lists. In *The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, pages 962–966, 2011. doi: 10.1109/ECTICON.2011.5948002.

Guangzhi Sun, Chao Zhang, and Philip C. Woodland. Tree-constrained pointer generator for end-to-end contextual speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 780–787, 2021. doi: 10.1109/ASRU51503.2021.9687915.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating Gender Bias in Natural Language Processing: Literature Review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1159. URL `https://www.aclweb.org/anthology/P19-1159`.

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`.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), pages 2818–2826, Las Vegas, Nevada, United States, jun 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.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. Multilingual Neural Machine Translation with Knowledge Distillation. In *Proceedings of International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, United States, 2019. URL `https://openreview.net/forum?id=S1gUsoR9YX`.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. Improving speech translation by understanding and learning from the auxiliary text translation task. 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 4252–4261, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.328. URL `https://aclanthology.org/2021.acl-long.328`.

Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. Unified speech-text pre-training for speech translation and recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1488–1499, Dublin, Ireland, May 2022. Association for Computational Linguistics.

doi: 10.18653/v1/2022.acl-long.105. URL `https://aclanthology.org/2022.acl-long.105`.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2020. URL `https://arxiv.org/abs/2009.06732`.

Mildred C. Templin. *Certain Language Skills in Children: Their Development and Interrelationships.* University of Minnesota Press, 1957.

Jörg Tiedemann. OPUS – Parallel Corpora for Everyone. *Baltic Journal of Modern Computing*, page 384, 2016. ISSN 2255-8942. Special Issue: Proceedings of the 19th Annual Conference of the European Association of Machine Translation (EAMT).

Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. A comparison of techniques for language model integration in encoder-decoder speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 369–375, 2018. doi: 10.1109/SLT.2018.8639038.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. Facebook ai's wmt21 news translation task submission. In *Proceedings of WMT*, 2021.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. Efficient speech translation with dynamic latent perceivers. *ArXiv*, abs/2210.16264, 2022a.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In *Proceedings of Interspeech 2022*, pages 106–110, 2022b. doi: 10.21437/Interspeech.2022-59.

Amos Tversky and Daniel Kahneman. Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 5(2):207–232, 1973.

Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. Neural machine translation incorporating named entity. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1274`.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. Getting Gender Right in Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1334. URL `https://www.aclweb.org/anthology/D18-1334`.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. Lost in Translation: Loss and Decay of Linguistic Richness in Machine Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 222–232, Dublin, Ireland, August 2019. European Association for Machine Translation. URL `https://www.aclweb.org/anthology/W19-6622`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5998–6008, Long Beach, California, December 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf`.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1117.

Piyush Vyas, Anastasia Kuznetsova, and Donald S. Williamson. Optimally Encoding Inductive Biases into the Transformer Improves End-to-End Speech Translation. In *Proc. Interspeech 2021*, pages 2287–2291, 2021. doi: 10.21437/Interspeech.2021-2007.

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.

Abraham Wald. Statistical Decision Functions. *The Annals of Mathematical Statistics*, 20(2):165 – 205, 1949. doi: 10.1214/aoms/1177730030. URL `https://doi.org/10.1214/aoms/1177730030`.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France, May 2020a. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.517`.

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 2020b. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.aacl-demo.6`.

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 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.80. URL `https://aclanthology.org/2021.acl-long.80`.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. Exploiting Cross-Sentence Context for Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1301.

Pengwei Wang, Xin Ye, Xiaohuan Zhou, Jinghui Xie, and Hao Wang. Speech2slot: An end-to-end knowledge-based slot filling from speech, 2021b. URL `https://arxiv.org/abs/2105.04719`.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-Attention with Linear Complexity, 2020c.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Jena D. Hwang, Claire Bonial, Aous Mansouri Jinho Choi, Maha Foster, Abdel aati Hawwary, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and Ann Houston. OntoNotes Release 5.0, 2012. URL `https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf`.

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.

Ian Williams, Anjuli Kannan, Petar Aleksic, David Rybach, and Tara Sainath. Contextual Speech Recognition in End-to-end Neural Network Systems Using Beam Search. In *Proceedings of Interspeech 2018*, pages 2227–2231, 2018. doi: 10.21437/Interspeech.2018-2416.

D.Randall Wilson and Tony R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10): 1429–1451, 2003. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(03)00138-2. URL `https://www.sciencedirect.com/science/article/pii/S0893608003001382`.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S.

Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, Jan 2008. ISSN 0219-3116. doi: 10.1007/s10115-007-0114-2. URL `https://doi.org/10.1007/s10115-007-0114-2`.

Shufang Xie, Yingce Xia, Lijun Wu, Yiqing Huang, Yang Fan, and Tao Qin. End-to-end entity-aware neural machine translation. *Machine Learning*, 111(3):1181–1203, March 2022. ISSN 0885-6125. doi: 10.1007/s10994-021-06073-9. URL `https://doi.org/10.1007/s10994-021-06073-9`.

Deyi Xiong and Min Zhang. A topic-based coherence model for statistical machine translation. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, page 977–983. AAAI Press, 2013.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. 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 2619–2630, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.204. URL `https://aclanthology.org/2021.acl-long.204`.

Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. End-to-End Named Entity Recognition from English Speech. In *Proceedings of Interspeech 2020*, pages 4268–4272, 2020. doi: 10.21437/Interspeech.2020-2482. URL `http://dx.doi.org/10.21437/Interspeech.2020-2482`.

Sane Yagi. Studying style in simultaneous interpretation. *Meta*, 45(3): 520–547, 2000. doi: https://doi.org/10.7202/004626ar.

Rong Ye, Mingxuan Wang, and Lei Li. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.376. URL `https://aclanthology.org/2022.naacl-main.376`.

Marcely Zanon Boito, Laurent Besacier, Natalia Tomashenko, and Yannick Estève. A Study of Gender Impact in Self-supervised Models for Speech-to-Text Systems. In *Proceedings of Interspeech 2022*, pages 1278–1282, 2022. doi: 10.21437/Interspeech.2022-353.

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.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. Adaptive Feature Selection for End-to-End Speech Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.230. URL `https://www.aclweb.org/anthology/2020.findings-emnlp.230`.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. Beyond sentence-level end-to-end speech translation: Context helps. 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 2566–2578, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.200. URL `https://aclanthology.org/2021.acl-long.200`.

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. Association for Computational Linguistics. doi: 10.18653/v1/D18-1049.

Shucong Zhang, Erfan Loweimi, Yumo Xu, Peter Bell, and Steve Renals. Trainable Dynamic Subsampling for End-to-End Speech Recognition. In *Proc. Interspeech 2019*, pages 1413–1417, 2019. doi: 10.21437/Interspeech.2019-2778.

Weitai Zhang, Zhongyi Ye, Haitao Tang, Xiaoxi Li, Xinyuan Zhou, Jing Yang, Jianwei Cui, Pan Deng, Mohan Shi, Yifan Song, Dan Liu, Junhua Liu, and Lirong Dai. The USTC-NELSLIP offline speech translation systems for IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 198–207, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.15. URL `https://aclanthology.org/2022.iwslt-1.15`.

Ziqiang Zhang and Junyi Ao. The YiTrans speech translation system for IWSLT 2022 offline shared task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 158–168, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.iwslt-1.11. URL `https://aclanthology.org/2022.iwslt-1.11`.

Ding Zhao, Tara N. Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. Shallow-Fusion End-to-End Contextual Biasing. In *Proceedings of Interspeech 2019*, pages 1418–1422, 2019. doi: 10.21437/Interspeech.2019-1209.

Jinming Zhao, Hao Yang, Gholamreza Haffari, and Ehsan Shareghi. M-Adapter: Modality Adaptation for End-to-End Speech-to-Text Translation. In *Proc. Interspeech 2022*, pages 111–115, 2022. doi: 10.21437/Interspeech.2022-592.

Yang Zhao, Lu Xiang, Junnan Zhu, Jiajun Zhang, Yu Zhou, and Chengqing Zong. Knowledge Graph Enhanced Neural Machine Translation via Multi-task Learning on Sub-entity Granularity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4495–4505, Online, December 2020a. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.397. URL `https://www.aclweb.org/anthology/2020.coling-main.397`.

Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. Knowledge Graphs Enhanced Neural Machine Translation. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4039–4045. International Joint Conferences on Artificial Intelligence Organization, 7 2020b. doi: 10.24963/ijcai.2020/559. URL `https://doi.org/10.24963/ijcai.2020/559`.

Lin Zheng, Chong Wang, and Lingpeng Kong. Linear complexity randomized self-attention mechanism. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27011–27041.

PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zheng22b.html.

Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett. Using Synthetic Audio to Improve the Recognition of Out-of-Vocabulary Words in End-to-End Asr Systems. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5674–5678, 2021. doi: 10.1109/ICASSP39728.2021.9414778.

Leiying Zhou, Wenjie Lu, Jie Zhou, Kui Meng, and Gongshen Liu. Incorporating Named Entity Information into Neural Machine Translation. In Xiaodan Zhu, Min Zhang, Yu Hong, and Ruifang He, editors, *Natural Language Processing and Chinese Computing*, pages 391–402, Cham, 2020. Springer International Publishing. ISBN 978-3-030-60450-9.

Lal Zimman. Transgender language, transgender moment: Toward a trans linguistics. In Kira Hall and Rusty Barrett, editors, *The Oxford Handbook of Language and Sexuality*. 2020. doi: 10.1093/oxfordhb/9780190212926.013.45.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual Data Augmentation for Mitigating Gender Stereotypes in Languages with Rich Morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1161. URL https://www.aclweb.org/anthology/P19-1161.